目　　录

⓿产品简介

❶实验平台

❷G6W 仿真器

❸G2K 仿真板（选件）

❹软件安装

❺硬件安装

❻实验例程（MCS51）

第一节 “验证式”实验例程

第二节 “模仿式” 实验例程

第三节 “探索式” 实验例程

推荐资料：

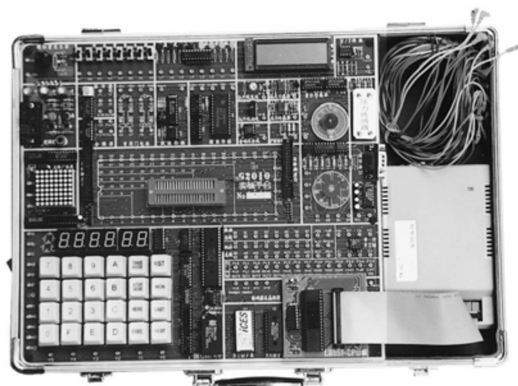
《MCS51-51 系列单片机实用接口技术》李华主编
北京航空航天大学出版社
《单片机的 C 语言应用程序设计》马忠梅等编著
北京航空航天大学出版社
《液晶显示应用技术》郭强等编著
电子工业出版社
《单片机原理及接口技术学习辅导》朱定华主编
电子工业出版社

❶ 产品简介

随着社会对人才素质要求的不断提高，同时也随着国家对教育投入的不断加大，向学生提供高性能的实验/开发设备成为必要和可能。根据“世行贷款高等教育项目”招标书中技术指标的要求，根据本公司对市场的调研，参考了国外先进教育设备的设计方法，研制了SICElab『赛思』开放式综合实验/仿真系统。

SICElab『赛思』开放式综合实验/仿真系统全面支持MCS51系列、MCS96系列、8086系列的实验仿真开发。同时还支持MICROCHIP公司的PIC系列CPU仿真、开发。

● 0.1 系统组成



SICElab『赛思』开放式综合实验/仿真系统由仿真器、实验平台、软件、开关电源组成。

传统的实验仪：

把仿真器和实验模块合二为一设计，实验模块作为仿真器的扩展部分,采用“单板式”方式进行实验，这使得仿真器仿真特性和实验方式真实性、灵活性都受到了限制。

我们知道单片机完整的开发过程为：确立方案 → 制作硬件 → 调测软件 → 固化代码→脱机运行。很显然，传统实验仪由于硬件的封闭性，是无法进行“确立方案”、“制作硬件”、“固化代码”、“脱机运行”的实验，其实验步骤与实际环境有较大差别。
另外，由于实验模块作为仿真器的扩展部分，占用了仿真器的资源，其仿真特性也大打折扣，事实上无法满足学生毕业设计、电子竞赛、教师科研所需。

『赛思』实验仪:

SICElab『赛思』开放式综合实验/仿真系统拼弃了传统实验仪的“单板式”设计方法，而采用了符合单片机开发过程的“仿真式”组合设计思想，使得所有的实验模块及CPU资源均全力对用户开放，从而充分满足“验证式”→“模仿式”→“探索式”→“开发式”的由浅入深的各种实验要求。并且，实验平台作为一个独立的目标系统，能让用户进行脱机验证实验结果的实验，从而，使实验步骤与实际开发环境完全一致，学以致用。同时，仿真工具作为一个可独立使用的仿真器，从而，又可满足学生毕业设计、电子竞赛、教师科研所需。达到一机多用之目的。

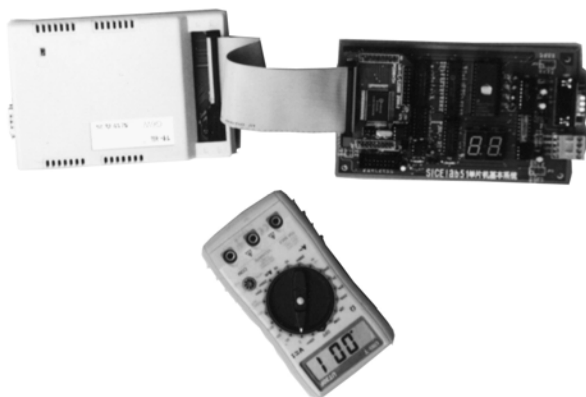
0．1．0　仿真器

由于SICElab『赛思』开放式综合实验/仿真系统的“仿真器”与“实验平台”可分离使用，故原则上可配备任何品牌的仿真工具。由于本手册是围绕南京伟福实业有限公司生产的仿真器进行编写，所以，选用该公司产品成为理所当然。

由于伟福公司所有的产品使用同一个大集成软件环境，统一的界面，包含一个项目管理器，一个功能强大的编辑器，汇编Make、build和调试工具并提供一个与第三方编译器的接口。由于风格统一，所以这就意味着可为SICElab『赛思』开放式综合实验/仿真系统选配伟福公司不同档次的仿真器，以满足各种不同场合需求，却并不影响整批产品的实验方案一致性。

作为仿真工具时，仿真器可与实验仪完全脱离

以下是伟福G6的性能指标：



☑ 仿真器采用XILINX高密度大规模可编程逻辑芯片设计，体积小、功能强，可靠性高、抗干扰能力强、性能好。

☑ 真正的双CPU架构，控制CPU在仿真器内，仿真CPU直接嵌入你的用户板微控制器插座，减少了中间环节，可靠性高、仿真频率倍增。

☑ 全透明，全实时在线仿真不占任何资源，包括仿真存储器，中断信号，8M地址空间。

☑ 128KB仿真RAM，能仿真超大容量CPU（如W78E438）。仿真频率最高达40MHZ，

☑ 串口通讯COM_1、COM_2均可，支持鼠标操作，通讯速率11，5200（每秒10K）与并口速度相当

☑ 该机具有最先进的静态硬件调测功能，将动态执行的指令静态化，配合逻辑笔和电压表，即可迅速查出硬件连线逻辑错误。

☑ 128硬件断点，绝对不受任何条件限制；夭折功能，在全速实时运行时从PC机打CTRL-C即可中断仿真器运行返回当前监控，并绝对正确地保持地址和数据，此功能对于查明死循环，程度走飞现象极为有效。

☑ 先进的硬件测试仪：可静态地设置数据/地址、ALE、PSEN、BHE、RD、WR，仅需逻辑笔或万用表即可测量非高既低电平，迅速定位硬故障。

以下是E2000的性能指标：

☑ 具备G6W的所有功能。

☑ 最高频率提高至100MHZ

☑ 逻辑分析仪：逻辑分析仪由交互式软件菜单窗口对系统硬件的逻辑或程序进行同步实时采样，并实时在线调试分析，采集深度32K，最高时基采集频率达20M，40路波形可精确实时反映用户程序运行时的历史时间。 配有三种触发方式：内外触发方式，静态触发方式，条件触发

方式 。系统在使用逻辑分析仪时，除普通的单步运行、键盘断点运行、全速硬件断点运行外，还可实现各种条件组合断点如：数据\地址、外部控制信号、CPU 内部控制信号、程序区间断点等，由于逻辑分析仪可以直接对程序的执行结果分析， 因此极大地便利了程序的调试。

☑ 逻辑跟踪器：追踪功能以总线周期为单位实时纪录仿真过程中 CPU 发生的总线事件，其触发条件方式同逻辑分析仪。追踪窗口在仿真停止时可收集显示跟踪 CPU 指令记忆信息，可以以总线反汇编码模式、源程序模式对应显示追踪结果。屏幕窗口显示波形最多追踪记忆指令 2048 条，并通过仿真器的断点、单步，全速运行或各种条件组合断点来完成追踪功能。总线跟踪程序的运行轨道。可以统计软件运行时间。

☑ 影子存储器：允许用户在实时运行时，观察变量。

☑ 无条件硬断点：可在任意地址、设置任意个硬件断点，并支持地址、数据、外部信号、事件等断点，程序区间断点及其各种方式组合。

☑ 程序时效分析：统计每个函数，每条指令的运行时间占整个程序运行时间的百分比。

☑ 数据时效分析：统计每个变量，每个存贮单元访问次数占整个程序访问次数的百分比。

☑ 事件触发器：可以设置地址条件、数据条件、控制条件、外部信号条件以及他们任意组合。事件触发器可以控制仿真器运行，同时也可以控制逻辑分析仪、跟踪器起停。

☑ 记时器：统计指定程序的运行时间，精度为 50ns。

☑ 代码覆盖：在用户程序运行时可以观察程序的执行情况。

☑ 波形发生器： 8 通道、每通道理３２K 深度，２０MHz 频率。

０．１．１　实验平台

新型实用模块：

（1）LCD 液晶实验　（2）点阵 LED 广告屏　（3）DS12887 实时钟　（4）红外线发送、接收
（5）直流电机恒速　（6）电子琴模拟实验　（7）串行ROM/I²C ROM（8）步进电机变速

传感器实验：

（1）温度传感器　（2）压力传感器　（3）霍尔传感器　（4）红外传感器

传统实验模块：

（1）模数转换 A/D0809　（2）数模转换 D/A0832　（3）8155 控制键显　（4）V/F 转换 LM331
（5）串口通讯 MAX232　（6）音响实验 LM386　（7）EPROM27C256 扩展（8）RAM6264 扩展
（9）微型打印机接口　（10）PWM 模块

通用实验模块：

（1）模拟信号发生器　（2）开关量发生器　（3）发光二极管组　（4）信号发生器
（5）74LS138 译码器　（6）分频器电路　（7）LED6 位数码管　（8）20 个键盘组
（9）逻辑笔　（10）常用门电路

自由实验模块：

由 DIP40 锁紧插座及 240 个插孔组成，CPU 所有信号均以插孔方式引出，还设计了常用门电路、晶振源、电源插孔等，可以完成以上实验模块的组合实验以及由实验者自行命题和新器件、新方案的实验，使得实验方式和内容不受限制。

扩展实验内容：

☙ 利用自由实验区可进行以下扩展实验：

（1）8279 键盘显示接口　（2）8255I/O 口扩展　（3）8253 可编程计数器
（4）8251 可编程通信接口　（5）8259 中断优先级管理器　（6）8237DMA 数据传送
（7）并口转串 74LS165　（8）串口转并口 74LS164　（9）并行I/O口扩展实验（10）AT89C51 最小系统　（11）I^2C总线 24C01　（12）串行 93C46
（13）GAL16V8 实验　（14）AT89C2051 实验　（15）FLASH29C256 实验
（16）译码器实验　（17）多个外中断实验

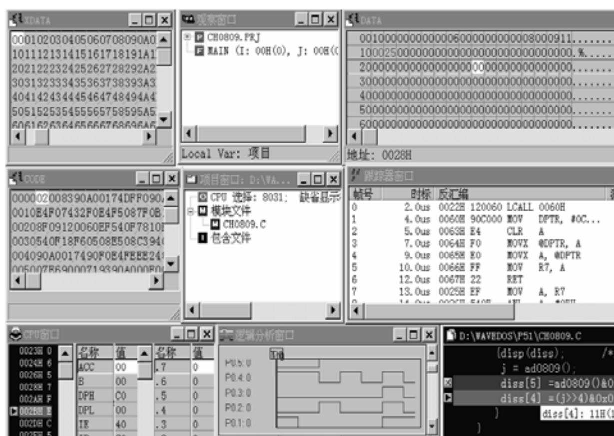
☙ 选配 G2000-通信扩展实验板，可进行以下实验：

（1）485 远程通信　（2）无线电通信 NRF40　（3）并口通信
（4）MODEN 通信　（5）嵌总线通信　（6）USB 通讯

模块自检接口：

“实验连线”对于学生理解实验内容的本质，提高动手能力的培养十分必要。然而，对实验室管理员而言，却是工作量倍增。试想，若对所有设备的所有实验模块进行一次全面的检测，将需连多少根线？工作量有多大？很显然，“模块自检接口”的自检功能十分必要。本产品的自检监控能对所有模块进行检测，管理员仅需在键盘上输入各自检程序的代码即可。

0．1．2　软件支持



DOS/WINDOWS 双平台。

WINDOWS 版本功能强大。中/英文界面可在线选择，用户源程序的大小不再有任何限制，支持 ASM、C、PLM 语言混合编程，源文本调试，具有项目管理功能，为用户的资源共享，课题重组提供强有力的手段。支持点屏显示，用鼠标左键点一下源程序中的某一变量，即可显示该变量的数值。丰富的显示方式，多方位、动态地显示仿真的各个过程，使用极为便利。全集成化调试环境，支持错误定位，在线修改功能。

为了跟上形势，现在工程师需要掌握不同的项目管理器、编辑器、编译器。它们由不同的厂家开发，相互不兼容，使用不同的界面，学习使用很吃力。伟福 WINDOS 调试软件为你提供了一个全集成环境，统一的界面包含一个项目管理器，一个功能强劲的编辑器，汇编 Make,Bulid 和调试工具并提供一个与第三方编译器的接口。由于风格一致，从而大大节省时间和精力。

DOS 版集成调试软件具有汇编、编译、调试和软件模拟等功能。所有操作均可通过窗口和菜单的选择来完成。国际标准的 BORLAND 窗口操作系统，一次可开 10 个以上窗口，支持 ASM、PLM、C 语言多模块混合语言源程序调试，在线直接修改、编译、调试、连接、执行源程序，错误定位，一键触发，自动完成。支持鼠标。

0．1．3　开关电源

SICElab『赛思』开放式综合实验/仿真系统已配备了+5V/2A、+12V/1A、-12V/0.5A 的开关

电源,直接使用 220V 交流电源工作。

● 0.2 实验内容

分成“验证式”、“模仿式”、“探索式”、“开发式”四个部分。“验证式”实验是指初学者仅需连几根关键的线或不连线，按实验手册中实验步骤即可完成的实验；“模仿式” 实验是指根据实验范例，通过调整连线可以产生新的方案的实验；“探索式”实验是指需要实验者自己设计实验方案，自己搭建主体电路，利用实验平台的辅助电路并需要自行查找故障原因，自行设计程序的实验。实验书中提供的几个“探索式”实验已提供了方案、电路、和参考程序，目的是要起到示例作用。“开发式” 实验是指完全需要自行设计的实验，由实验者自行立题。**用户可根据各自情况选用实验内容，或完全自行设计实验内容。**

各实验例程均提供 ASM 语言、C 语言两套程序清单。以上实验已充分涉及单片机的原理、接口、传感器、自控原理等方面的内容。

● 0.3 实验方式

- 单板机实验方式: 配 G2K 仿真板，使用实验仪上键盘、LED 显示进行实验，可以满足无计算机条件下的实验所需。
- 外接仿真器方式： 用 PC 机联仿真器，再联实验仪进行仿真和实验。
- 软件模拟的方式： 无实验仪、无仿真器，仅在 PC 机上采用模拟软件进行实验。
- 仿真器独立方式： 用户自制目标板，利用仿真器进行仿真实验。
- 自检测实验方式： 利用自检插口和自检监控可进行演示实验，简单可靠。

● 0.4 支持器件

- 选用 Lab51CPU 板可支持 MCS51 系列 CPU 的实验；
- 选用 Lab96CPU 板可支持 MCS96 系列 CPU 的实验；
- 选用 Lab88CPU 板可支持 IN8086 系列 CPU 的实验；
- 选用 POD51 仿真头可支持以下 CPU 的仿真开发：

  INTEL 系列 8 位 CPU：8031/32/51/52、87C51/52、8751FA/FB/FC、8044、8344
  ATMEL 系列 8 位 CPU：89C51/52/54/55/2051/1051
  LG 系列 8 位 CPU：97C51/52、97C54/56/58/2051/1051
  WINDOWS 系列 CPU：78E51/54/58
  SST 系列 8 位 CPU：89C58、89C51、89C52、89C59/2051/1051
- 选用 POD96 仿真头可支持以下 CPU 的仿真开发：

  INTEL 系列 16 位 CPU：80C196KB/KC、87C196KB/KC、8094/95/96
- 选用 POD552/320/438 仿真头可支持以下 CPU 的仿真开发：

  WINDOW 公司的：78E438、78E538（PLCC84 封装）
  DALLAS 系列 8 位：80C320
  PHLIPS 系列 8 位 CPU：80C552、80C562、80C654、80C550、87C451
- 选用 POD-PIC 仿真头可支持以下 CPU 的仿真开发

  MICRCHIP 公司的：PIC16C5X、PIC16C5XA、PIC14CX。PIC16C71X、PIC16C8X、PIC16C92X

❶ 实验平台

传统的单片机实验仪，是把实验模块作为仿真器的一个部分进行一体化设计，即所谓的“单板式”设计方法。其在实验过程中并不涉及“仿真状态”（或称工作模式）这个重要的概念，亦不能进行开发式实验，调试的程序不能进行脱机运行。显然，这种实验仪的实验过程与实际的开发步骤存在较大的差距。另外，由于此类实验仪的仿真器，大多为单 CPU 架构，其仿真 RAM 空间被实验模块占用，所以，其仿真性能也不适合作为仿真工具所用。

SICElab『赛思』开放式综合实验/仿真系统采用“仿真式”设计方法，仿真器与实验平台分离，采用“仿真”方式进行实验，同时，允许进行脱机运行工作，所以，实验过程是与实际开发过程完全一致。仿真器使用的是双“CPU”架构方式，100%资源出让，100%实时，100%无条件硬件断点，可满足学生毕业设计，参加电子竞争，教师科研所需。

G2010+实验平台可根据教学实践的需要实现 MCS51/MCS96/8086/8088 单片机、微机原理与接口实验，并在硬件上预留了自主开发实验的空间。该平台对基本实验仪需少量连线就可进行，以减少学员工作量，同时也提供了一些需较多连线的扩展性实验以进一步锻炼学员的动手能力。

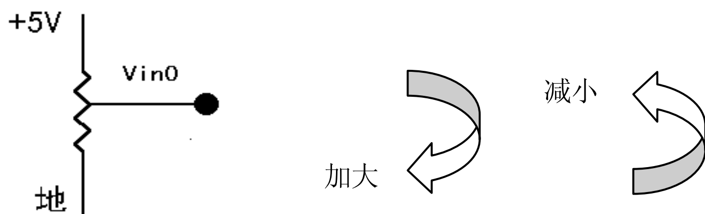
模块自检接口：“实验连线”对于学生理解实验内容的本质，提高动手能力的培养十分必要。然而，对实验室管理员而言，却是工作量倍增。试想，若对所有设备的所有实验模块进行一次全面的检测，将需连多少根线？工作量有多大？很显然，“模块自检接口”的自检功能十分必要。本产品的自检监控能对所有模块进行检测，管理员仅需在键盘上输入各自检程序的代码即可。

● 1.1 实验模块

G2010+实验平台有丰富的实验电路和灵活的组成方法。这些电路既可以和 Lab51CPU 板（51 系列）组合，以完成 MCS51 系列实验;也可以和 Lab96CPU 板（80C196 系列）组合，以完成 MCS96 系列实验;还可和 Lab88CPU 板相连,以完成 8086 系列实验。为了描述清楚，在此作统一的介绍。

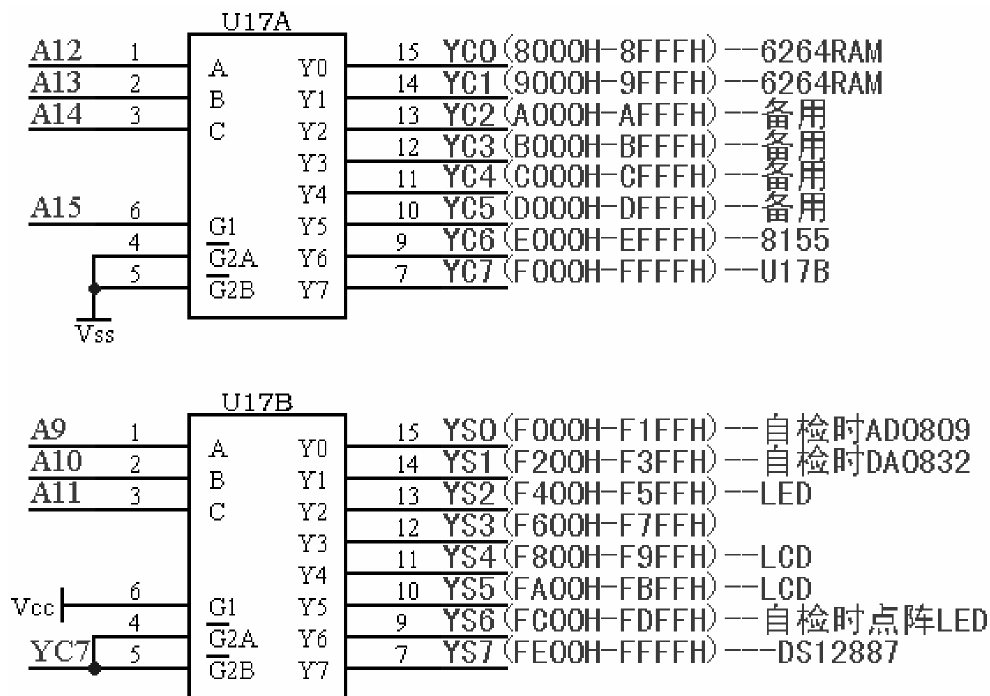
1.1.0 模拟信号发生器

电位器电路用于产生可变的模拟量。顺时钟旋转，电压值加大；反之，减小。



1.1.1 138 译码器

为了使得 MCS51、MCS96、8086 兼容实验，所以 ROM、RAM 同 64K 空间统一分配地址，程序空间占用前 32K（0000—7FFFH)，数据空间占用后 32K（8000H-0FFFFH)，使用两片 74LS138 译码器对后 32K 空间进行译码。其中：



EPROM27C256:（0000H-7FFFH）

YC0-YC1（8000H-9FFFH)：6264RAM

YC7（0F000H-0FFFFH)：U17 号 74LS138 选通

YC6（0E000H-0EFFFH)：8155

YC2（0A000H-0AFFFH)：备用

YC3（0B000H-0BFFFH)：备用

YS7（0FE00H-0FFFFH)：DALLAS12887

YS6（0FC00H-0FDFFH)：自检时的点阵 LED

YS5（0FA00H-0FBFFH)：LCD 液晶显示

YS4（0F800H-0F9FFH)：LCD 液晶显示

YS2（0F400H-0F5FFH)：LED 发光二极管

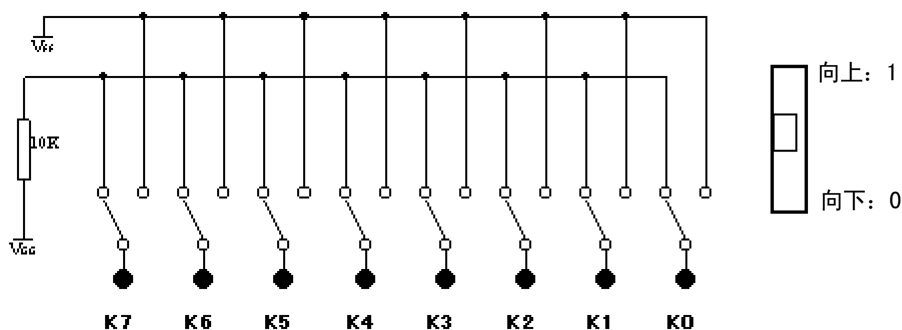
YS1（0F200H-0F3FFH)：自检时的 DA0832

YS0（0F000H-0F1FFH)：自检时的 AD0809

1.1.2 开关量发生器

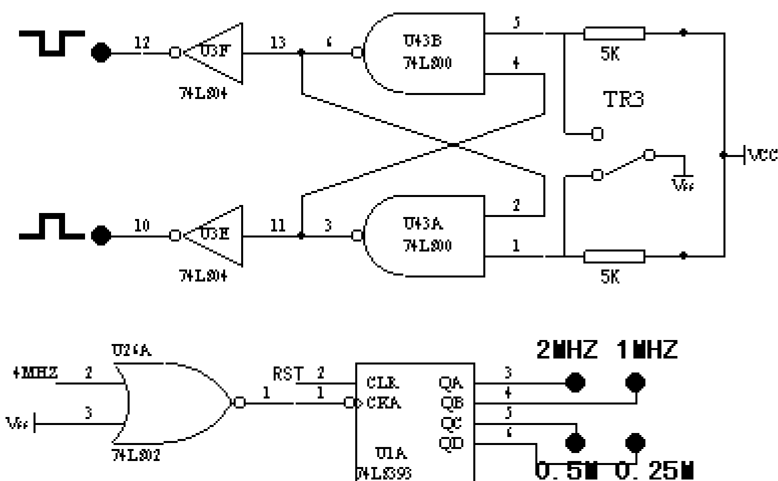
实验平台上有 8 只拨动开关 K0-K7 及相应的驱动电路，以产生“1”、“0”的逻辑电平。开

关向上拨相应插孔输出高电平为“1”，反之，输出低电平为“0”。



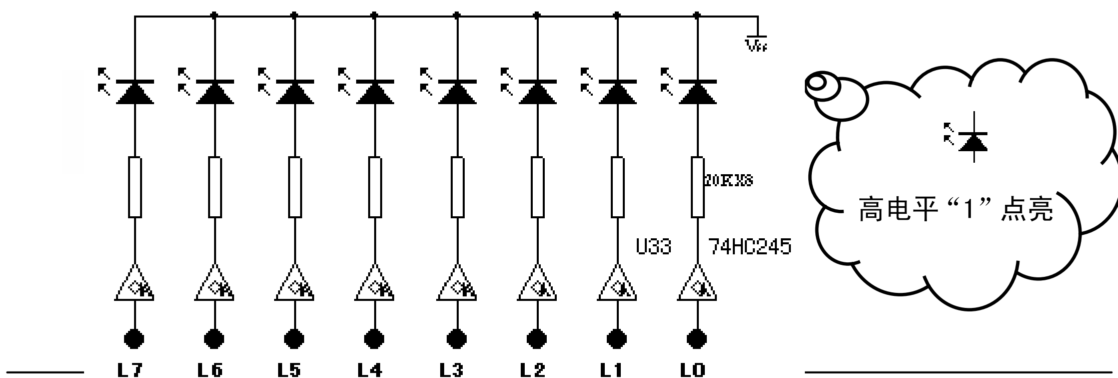
1.1.3 信号发生器

由 U3 的 74LS04、U43 的 74LS00 组成，每按一次带锁开关即产生一个单脉冲。



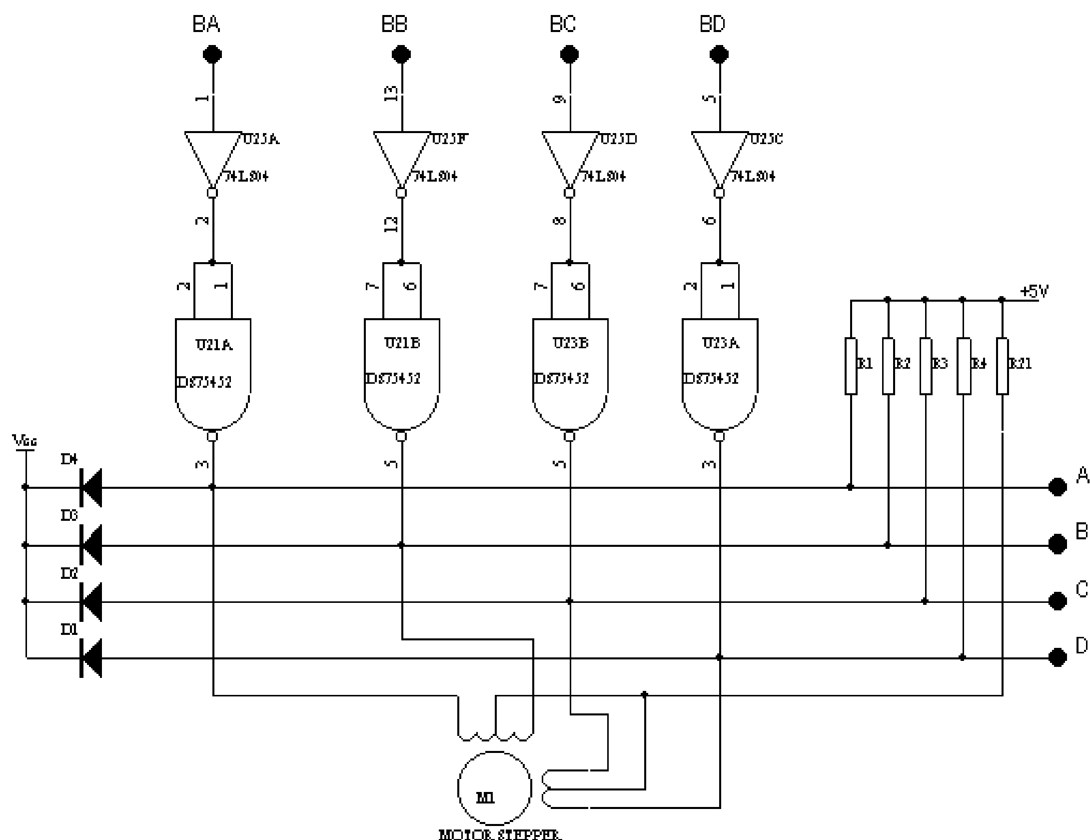
1.1.4 发光二极管组

实验平台上有 8 只发光二极管，由 U33 的 74HC245 驱动，以显示电平状态。**高电平“1” 点亮发光二极管。**

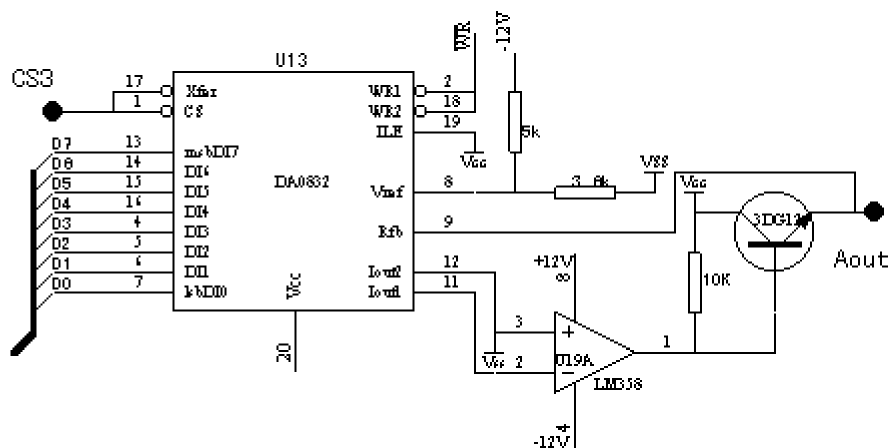


1.1.5 步进电机实验电路

G2010+实验平台选用的是四相步进电机，由 U25 的 74LS04 和 U21、U23 的 75452 驱动。

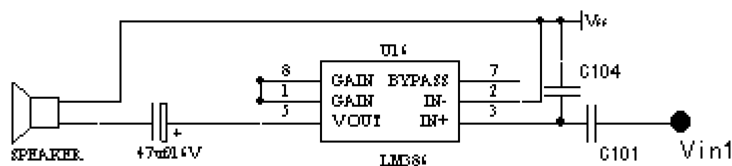


1.1.6 D/A0832 模块

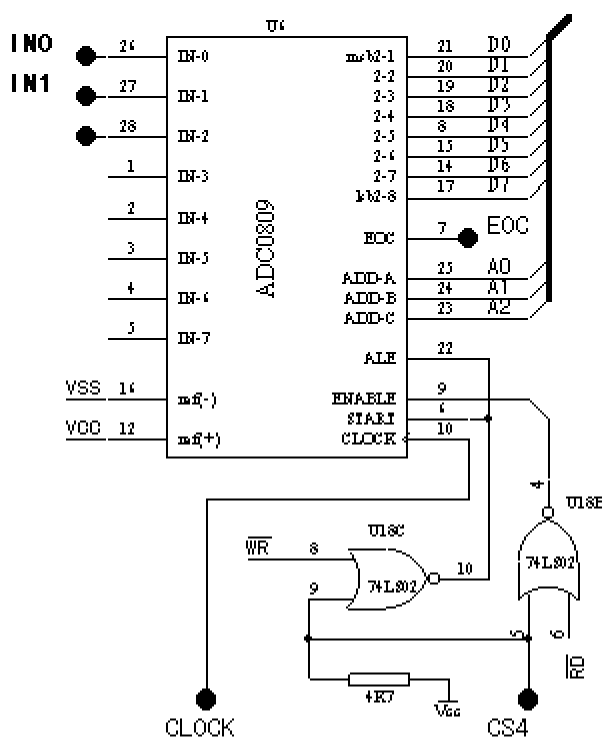


1.1.7 音响实验

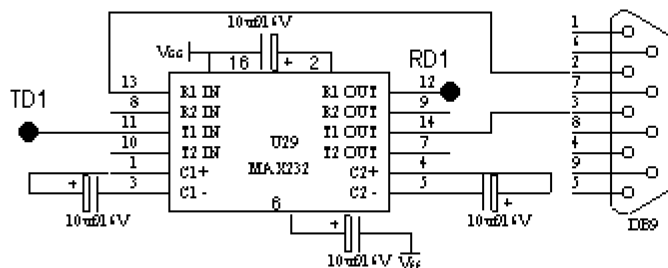
喇叭由 U16 的 LM386 驱动。



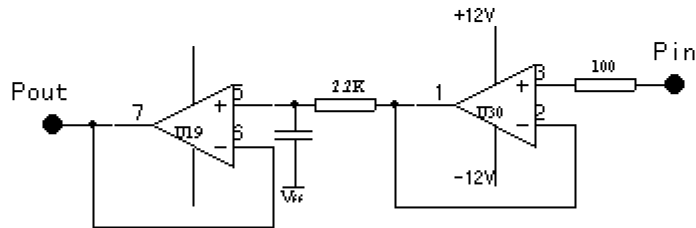
1.1.8 AD0809 模块



1.1.9 RS232 通讯模块

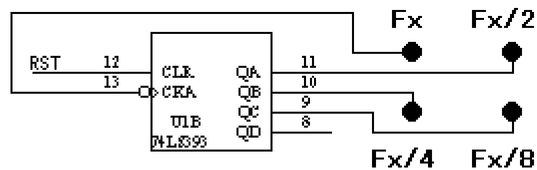


1.1.10 PWM 模块

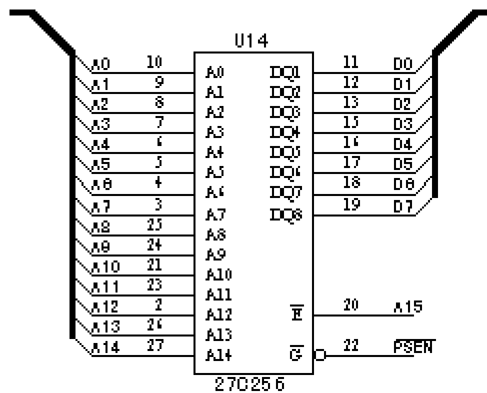


1.1.11 分频器模块

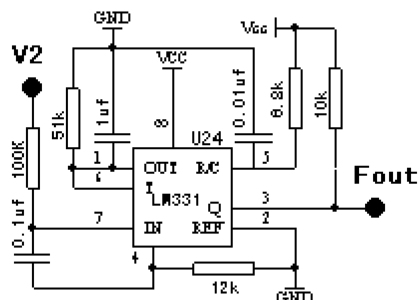
由 74LS74 的一组锁存器组成二分频的分频器，另一组的引脚均以插孔方式引出。如把 2D 孔与 2/Q 孔相连还可产生另一个二分频的分频器。



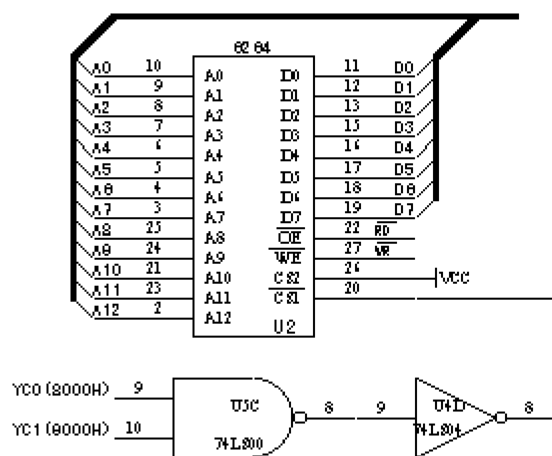
1.1.12 EPROM27256 扩展模块



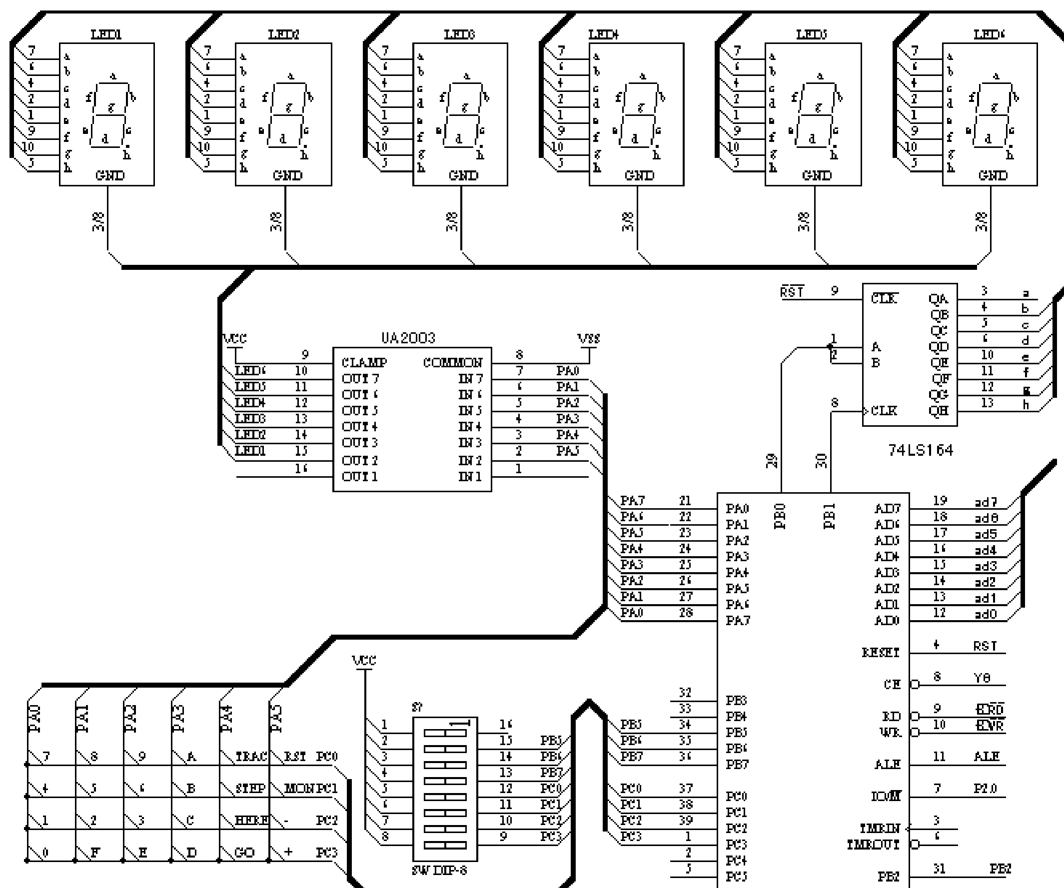
1.1.13 V/F 压频转换



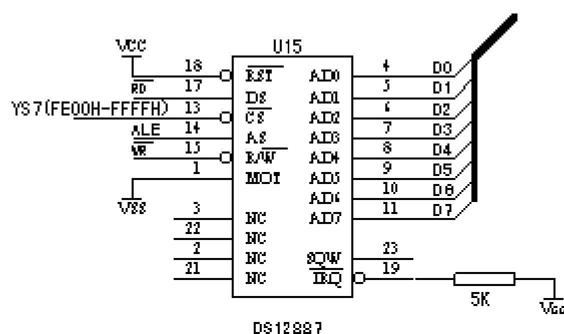
1.1.14 RAM6264 扩展模块



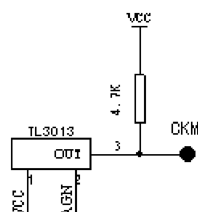
1.1.15 8155 键显模块



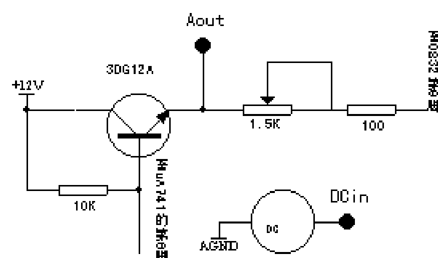
1.1.16 DALLAS12887 时钟模块



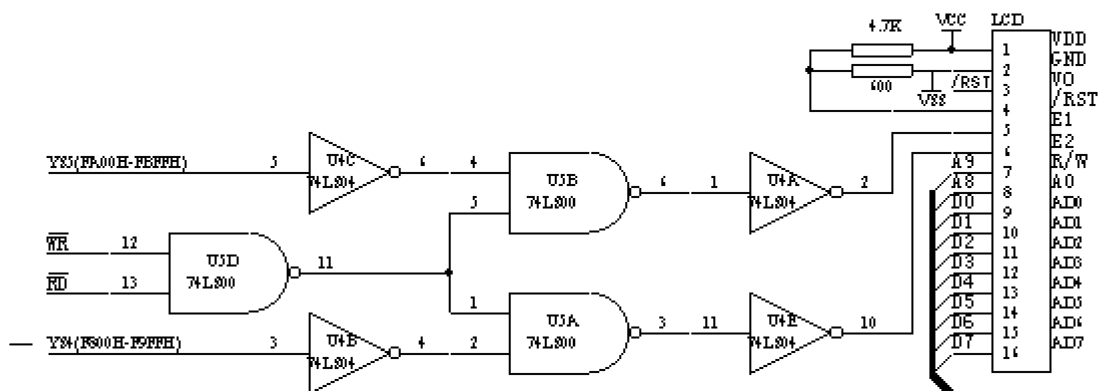
1.1.17 霍尔传感器



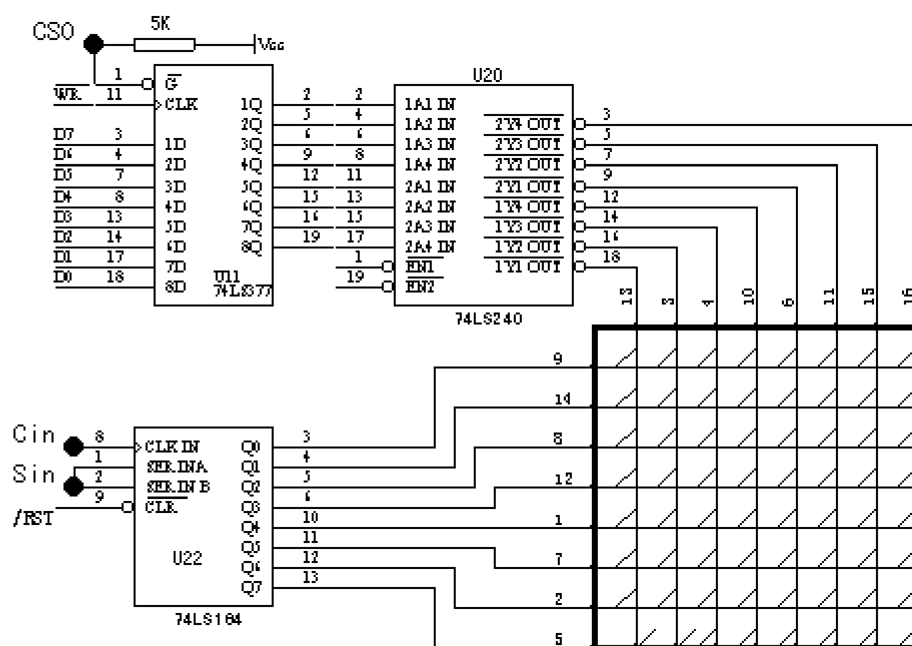
1.1.18 直流电机



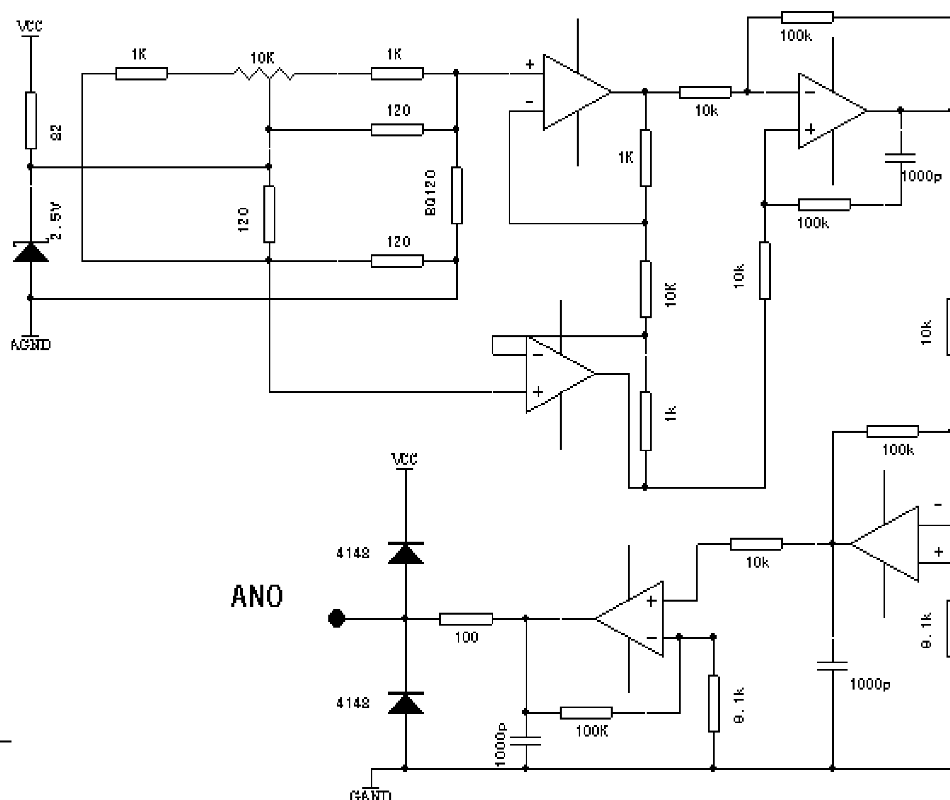
1.1.19 122X32LCD 液晶显示模块



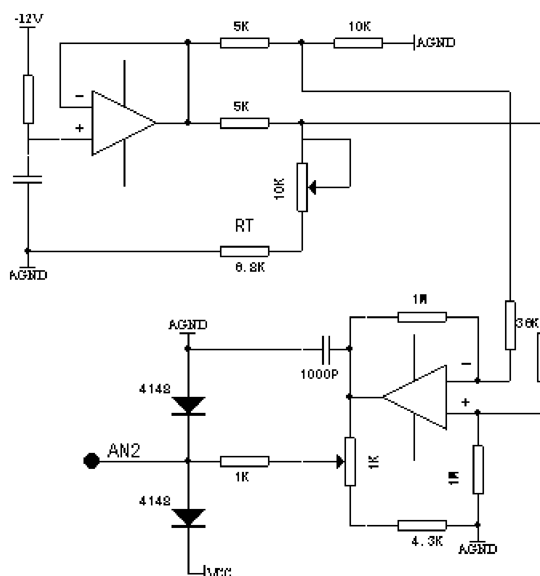
1.1.20　点阵 LED 模块



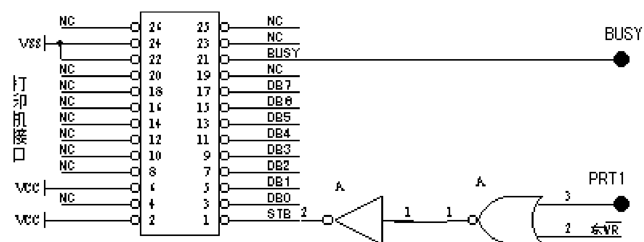
1.1.21　压力传感器



1.1.22　温度传感器

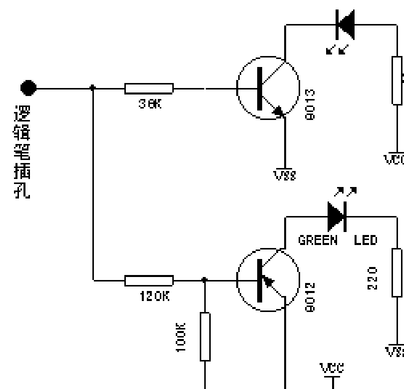


1.1.23　微型打印机接口

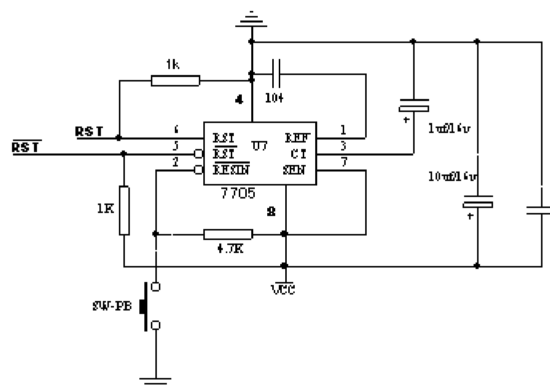


1.1.24　逻辑笔电路

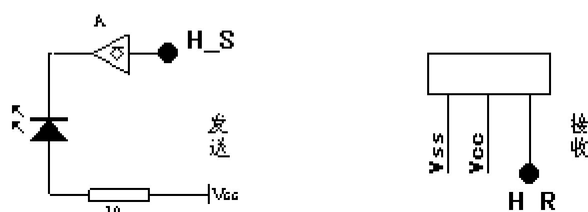
G2010 实验平台上有逻辑测量电路，用于测量各种电平，其中红灯亮表示高电平，绿灯亮表示低电平。如果两灯同时闪动，表示有脉冲信号；两灯都亮时，表示浮空（高阻态）。



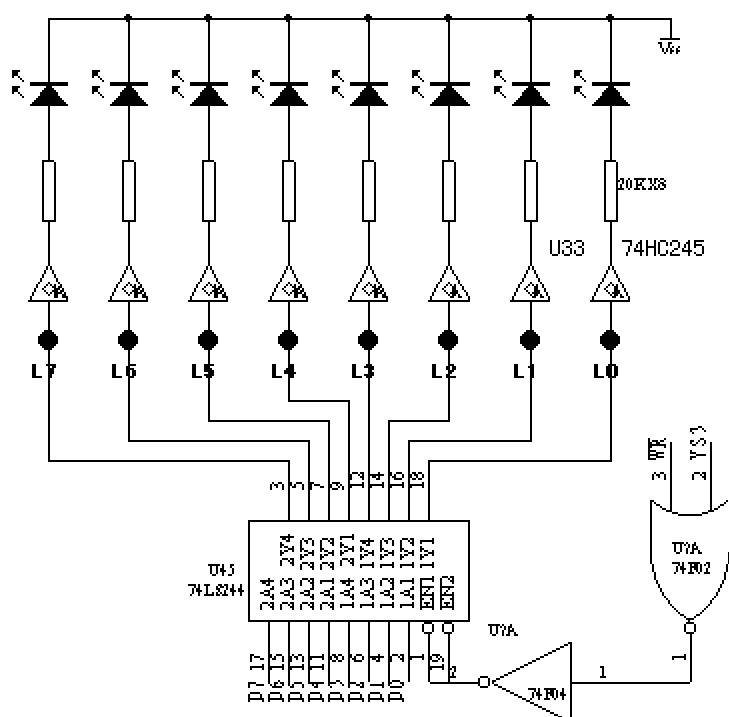
1.1.25 复位电路



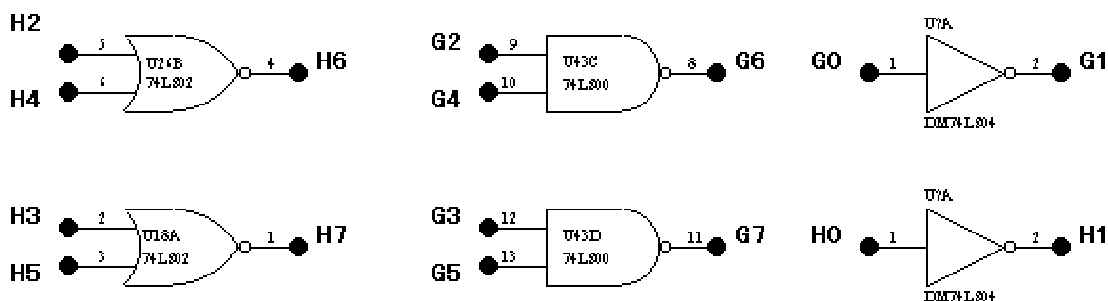
1.1.26 红外线发送/接收



1.1.27 LED 发光二极管总线驱动

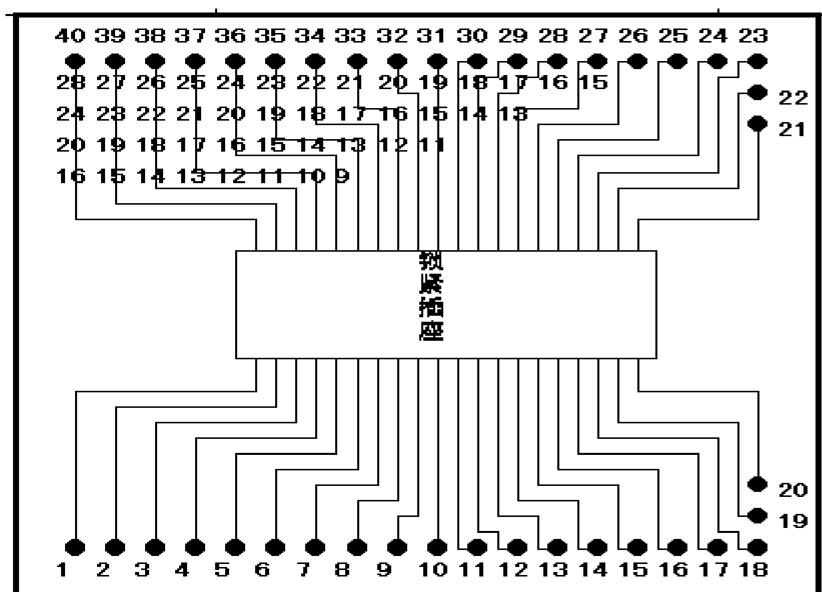


● 1.2 常用逻辑门电路

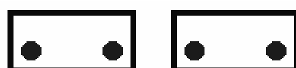


● 1.3 自由实验插座

G2010+实验平台设计了一个锁进插座,以供自开发实验用，插座全部引脚都被引出到相应的插孔，40 芯、32 芯、28 芯、24 芯、20 芯、16 芯、14 芯、8 芯通用，并按照各自的封装标明引脚号。利用这些插座，可对双列直插式的各种微机芯片进行实验。



● 1.4 跳线器选择：

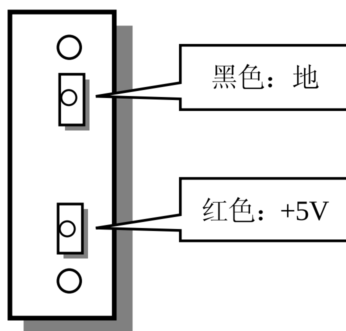


74LS240工作



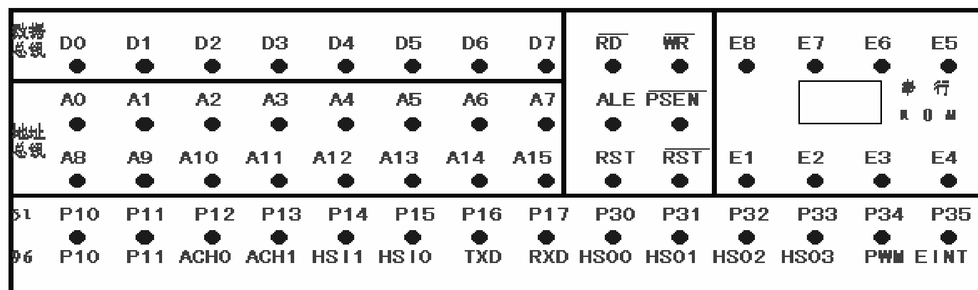
74LS240不工作

● 1.5 直流电源外引插座：



● 1.6 总线插孔：

G2010+实验平台的所有总线及控制信号均以插孔方式引出，以便进行开放式实验。



● 1.7 空间分配：

扩 展 模 块	资源分配(138 译码)
27C256	0000H-7FFFH
6264	(YC0,YC1)8000H-9FFFH
8155	(YC6)0E000H-0EFFFH
LCD 液晶显示	(YS4-YS5)0F800H-0FBFFH
DS12887	(YS7)0FE00H-0FFFFH
LED 二极管总线驱动	(YS2)0F400H-0F5FFH
自检时 AD0809	(YS0)0F000H-0F1FFH
自检时 DA0832	(YS1)0F200H-0F3FFH
自检时点阵 LED	（YS6）0FC00H-0FDFFH
自检时微型打印机	YC2（0A000H-0AFFFH）
备　用	(YC2)0A000H-0AFFFH
备　用	(YC3)0B000H-0BFFFH

❷ G6W 仿真器

南京伟福实业有限公司在过去的十年中先后推出了 G3、G5、G6、ICExplorer 系列仿真器。现在，伟福公司已成为 Microchip、PHILIPS、TI, WINBOND 等知名单片机生产商的签约合作伙伴，为他们的单片机提供仿真器支持。最新推出的“E2000”系列仿真器采用了超大规模定制芯片及专用仿真技术制造，集仿真器、逻辑分析仪、跟踪器、逻辑笔、波形发生器，影子存储器、记时器、程序时效分析、数据时效分析、事件触发器于一体的通用仿真器，将仿真器软、硬件提高到前了所未有的水平。

考虑到大专院校学生的实际使用情况，南京伟福实业有限公司为院校设计了 G6W 仿真器。G6W 仿真器，与 E2000 系列相比，在主要性能指标上相同或相近：支持仿真的器件一样多；DOS/WINDOW 双平台且使用环境完全一致；支持逻辑笔和静态测试,完全无条件硬断点；仿真 CPU 外挂，以减少噪声，提高仿真频率；在功能上减少逻辑分析仪，逻辑跟踪等。

● 2.0 G6W/G6S 型仿真器

仿真器型号	功　　　能
G6S	通用仿真器 （1-16 位，15M 总线速度） 硬件测试仪、DOS 版本
G6W	含 G6S 所有功能 WINDOWS 版本、DOS 版本双平台

G6W/G6S 型仿真器先进的特点如下：

- 主机+POD 组合，通过更换 POD，可以对各种 CPU 进行仿真

  对待不同的应用场合，用户往往会选择不同的 CPU，从而需要更换仿真器，伟福仿真软件 WINDOWS 版本支持本公司多种仿真器。支持多类 CPU 仿真。仿真器则采用主机+POD 组合，通过更换不同的 POD，可对各种不同类型的单片机进行仿真。为用户提供了一种灵活的多 CPU 仿真系统。

- DOS 版本，WINDOWS 版本双平台

  其中 WINDOWS 版本功能强大。中文界面，英文界面可任选，用户源程序大小不再有任何限制，支持 ASM，C，PLM 语言混合编程，具有项目管理功能，为用户的资源共享，课题重组提供强有力的手段。支持点屏显示，用鼠标左键点一下源程序中的某一变量，即可显示该变量的数值。有丰富的窗口显示方式，多方位，动态地显示仿真的各种过程，使用极为便利。

- 双工作模式

  （1） 软件模拟仿真（不要仿真器也能模拟仿真）。

  （2） 硬件仿真。

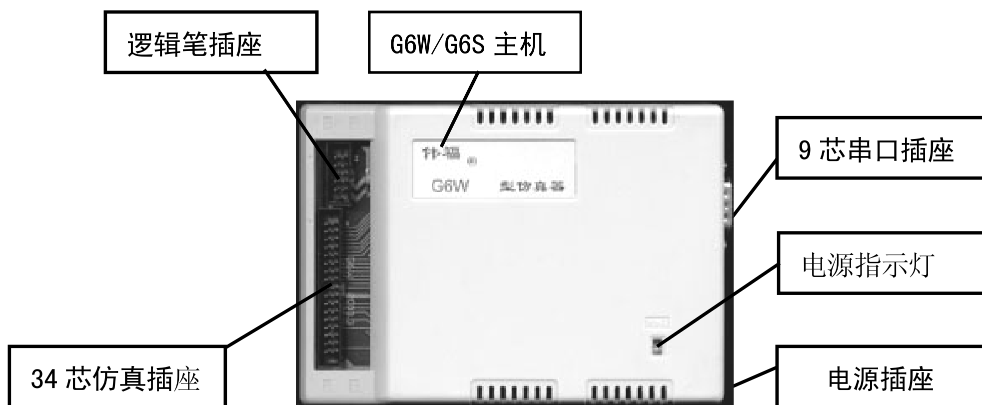
- 双 CPU 结构，100% 不占用户资源。全空间无条件硬断点，支持运行时间统计。
- 双集成环境

  编辑、编译、下载、调试全部集中在一个环境下。多种仿真器，多类 CPU 仿真全部集成在一个环境下。可仿真 51 系列，196 系列，8086 系列,PIC 系列，飞利蒲公司的 552、 DALLAS320，华邦 438 等 51 增强型 CPU。为了跟上形势，现在很多工程师需要面对和掌握不同和项目管理器、编辑器、编译器。他们由不同的厂家开发，相互不兼容，使用不同的界面。学习使用都很吃力。伟福 WINDOWS 调试软件为您提供了一个全集成环境，统一的界面，包含一个项目管理器，一个功能强大的编辑器，汇编 Make、Build 和调试工具并提供一个与第三方编译器的接口。由于风格统一，从而大大节省了您的精力和时间。

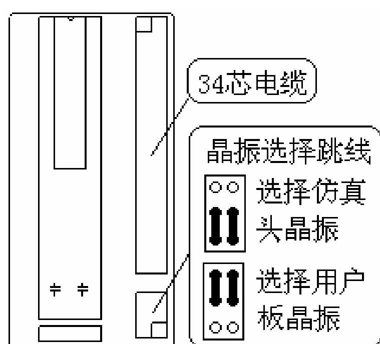
● 2.1 可配置仿真头

仿真头型号	可　仿　真　CPU
POD89C52	8X5X 系列（P0 口和 P2 口作为 IO 口用）
POD51	8X5X 系列 CPU（P0 口和 P2 口作为总线用）
POD2051	2051、1051 系列 CPU（需与 POD8952 配合使用）
POD552	Philips 80C552
POD196KC	196KC/KB
POD196MC	196MC/MD
POD16C5X	PIC16C52/54/55/56/57/58、PIC12C508/509
POD16C6X/7X	PIC16C61/62/63/64/65/67、PIC16C71/72/73/74
POD16C71X	PIC16C71X 全系列
POD16C8X	PIC16C83/84
POD8086	8088/8086CPU

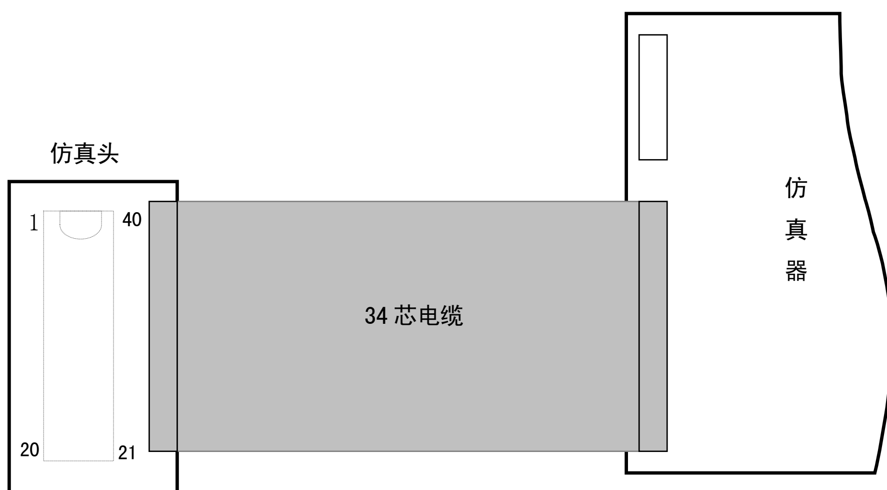
● 2.2 G6W/G6S 外形示意图



● 2.3 POD8051 仿真头

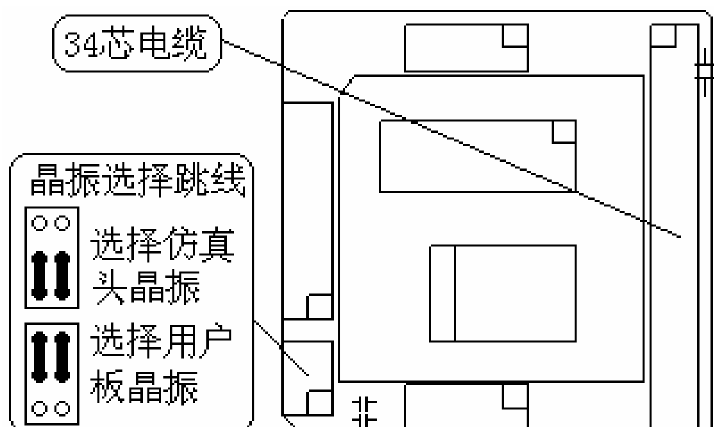


用于仿真 P0,P2 口做为总线工作方式的 8031/32,8051/52 系列及兼容的 89/87/97/78 系列单片机,可选配 44 脚 PLCC 封装的转接座,用于仿真 PLCC 封装的芯片。

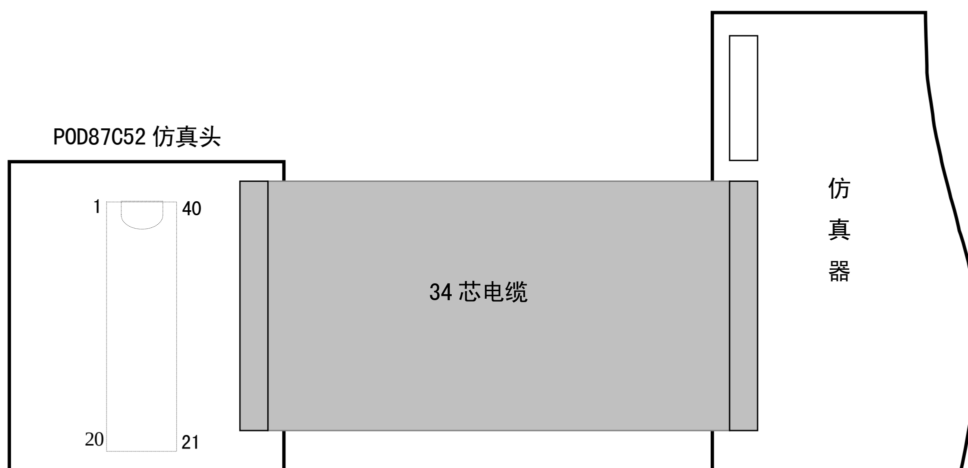


G6W / G6S 仿真器与 POD8051 连接图

● 2.4 POD89C52 仿真头



POD89C52 仿真头用于仿真 P0,P2 做为 I/O 口方式的 89/87/78/97C51/52/55/58/59 系列单片机,此种仿真头适用于 G6W/G6S 型仿真器. 可选配 44 脚 PLCC 封装的转接座,用于仿真 PLCC 封装的芯片. 选配 2051 转接可仿真 20 脚 DIP 封装的 89/97C1051/2051/4051

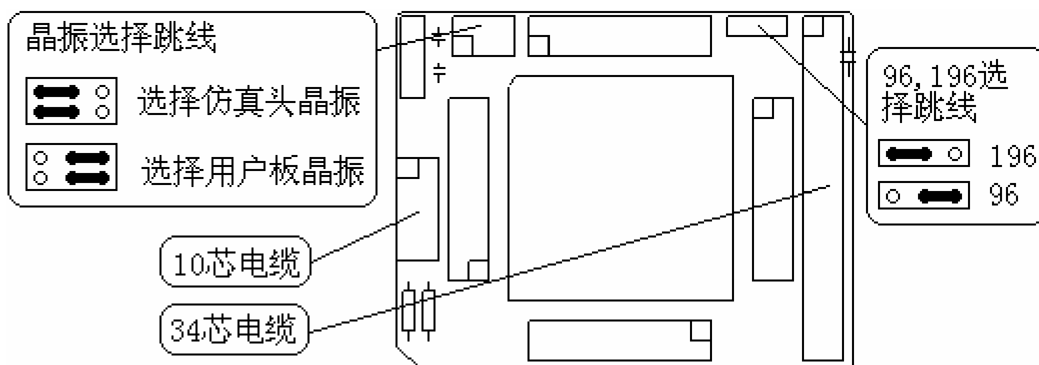


G6W / G6S 仿真器与 POD89C52 连接图

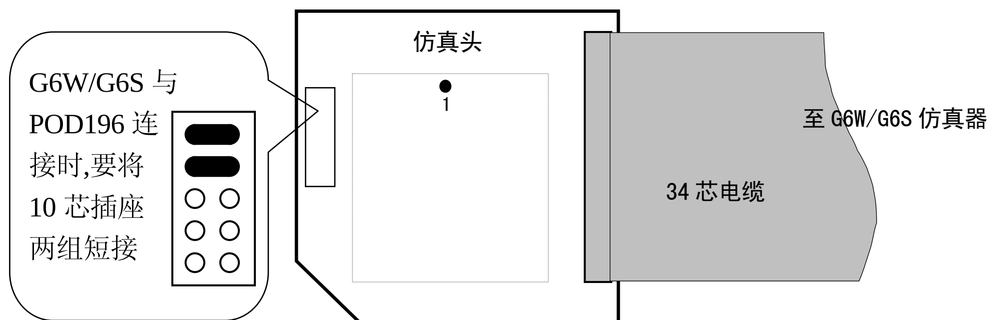
特别指出：

究竟使用 POD51 还是 POD89C52，不是看用户系统使用的 CPU 是 8051/52 还是 89C51/52，而是看 P0、P2 口作何用。如把 P0、P2 口作总线使用，即使使用的 CPU 为 89C51/52/55，也选用 POD51；如把 P0、P2 口作 I/O 口使用，则应选用 POD89C52。

● 2.5 POD196KB/KC 仿真头

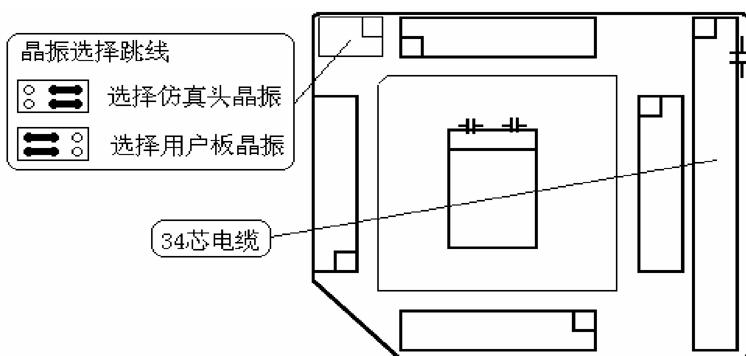


用于仿真 INTEL80C196KB/KC 单片机

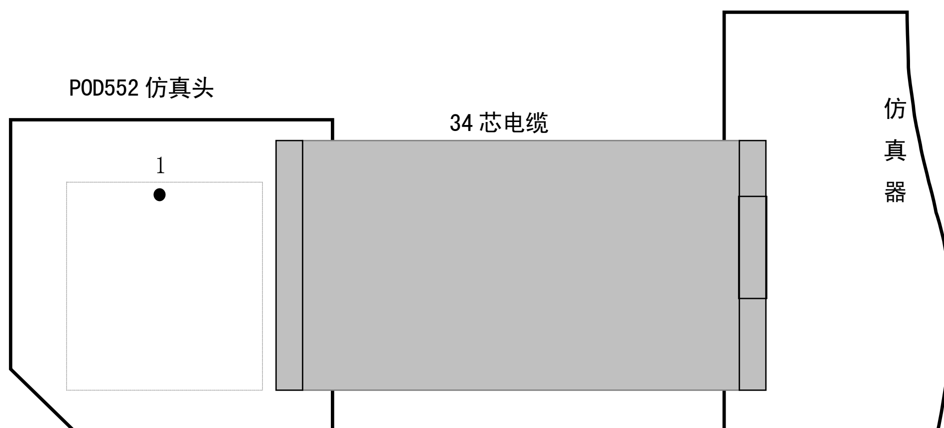


G6W / G6S 仿真器与 POD196KC 连接图

● 2.6 POD552 仿真头

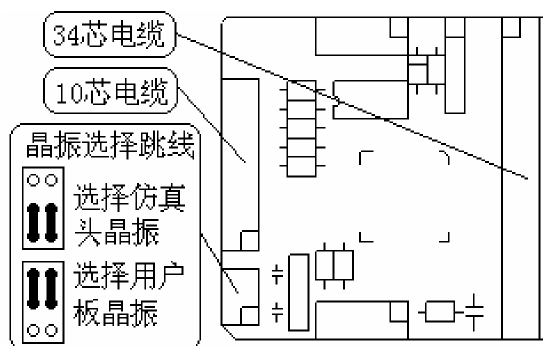


用于仿真 PHILIPS 公司的 80C552 单片机



G6W / G6S 仿真器与 POD552 连接图

● 2.7 POD16C5X 仿真头



用于仿真MICROCHIP公司的PIC16C5X系列单片机,可仿真芯片PIC16C52/54/55/56/57/58 PIC16C508/509,同时配有8脚/18脚/28脚三种DIP封装转接座,适用于不同的芯片.



🛠 更多信息请参阅《伟福仿真器使用手册》

🛠 网址：www.wave-cn.com

❸ G2K 仿真板

（选配件）

由于各个学校的实验条件、实验目的不尽相同，故专门设计生产了具备强劲键盘调试能力的 G2K 仿真板。G2K 仿真板与 G2010+实验平台配合，可满足无计算机条件下的实验所需。它也可和计算机串口联接，利用计算机联调，以满足单片机为非专业课、对仿真能力要求不强、需要廉价实验设备的单位和个人所需。

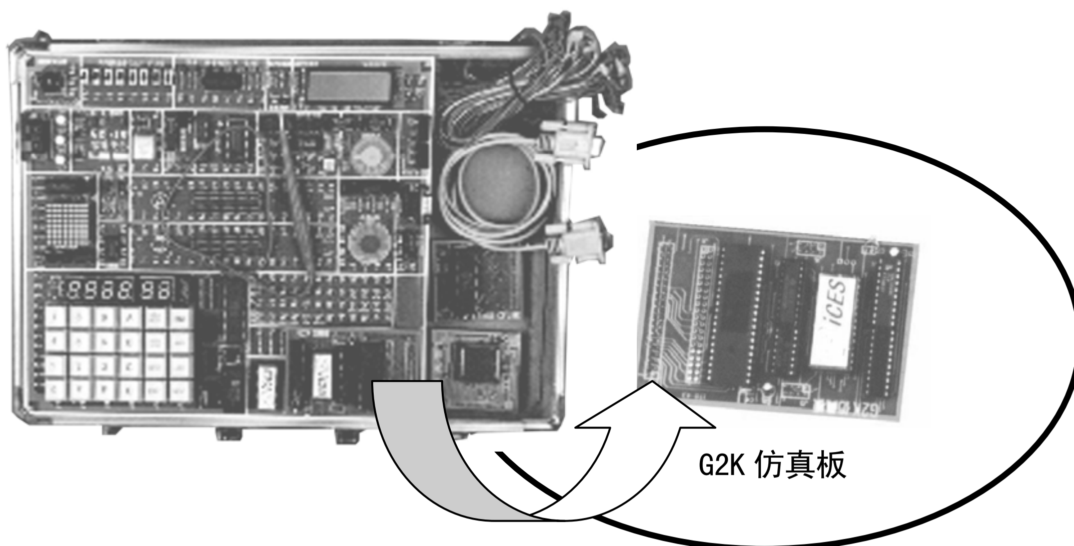
当 G2K 仿真板与 G2010+实验平台配合使用时，键盘、数码管、RS232 串口已被 G2K 仿真板占用，用户不能再使用。

无系统机时，应先用汇编语言写好程序，用人工方法或者利用ＰＣ机把汇编语言翻成机器码，再用实验仪上的键盘将机器码输入，然后用 Trace 键，Here 键，Step 键进行调试，用 Exec 键使程序运行。

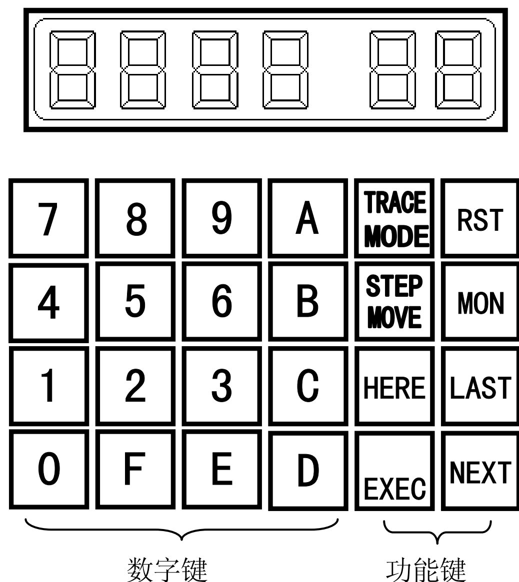
G2010+实验平台选配 G2K 仿真板后，将具备强劲的键盘调试能力。

● 3.1　G2K 仿真板与 G2010+实验平台的安装方法：

G2K 仿真板安装在 G2010+实验平台的 J1、J2、J3 插座上，如下图，通电后，在 LED 上显示出 CPU 的型号，如 8051。



● 3.2 键盘使用说明



本实验仪共有 6 位 LED 发光数码管, 左边 4 位为地址位，右边 2 位为数据位。

当地址位 4 位数均有数值时，表示地址是程序存储器或数据存储器的地址。当地址数码管只有 2 位时（右边 2 位发光，左边 2 位暗)，表示内部寄存器地址。当地址数码管只有 3 位时（最左面 1 位暗，第 2 位显示“0”，第 3 和 4 位是数字），表示是内部特殊寄存器（SFR）的地址。

最右边的两位是数据位，它表示的是左边显示的地址单元中的数据。

面板上共有 24 个小键盘,为了介绍和使用时查找方便,约定用方括号表示按键,例如,[RST]表示面板上的“RST”键。

3.2.0 [RST]键 [整机复位键]

整机复位键,复位后数码管的地址位显示“8051”字样(对于 MCS196 系列, 显示‘8096’),表示复位操作完成。复位后程序存储器和外部数据存储器中的内容不变，程序指针回到 0000处(8096 系列为 2080H)。CPU 内部寄存器复位后为单片机复位操作规定的值，即有的寄存器的数据不变, 有的寄存器中的数据被复位。

3.2.1 [0..F]键 [数字输入键]

数字键，用于输入 16 进制数。仿真器中的数据和地址均是用 16 进制表示。

3.2.2 [MON]键 [监控键]

它有两个作用：

用于选择上档和下档键

面板上有 2 个上下档的键[Trace/Mode]和[Step/Move]，当第一次按[MON]键时 LED 数码管最左侧的一只显示为“P”字符，表示选中上档，此时两个上档键 Trace 和 Setp 起作用。如果

再按一次[MON]键，在左侧的数码管显示为“’”（即只有右上角的一竖亮），则这两个键的下档Mode 和 Move 起作用。

表示某些操作已结束

某些操作的所有步骤完成后，按 MON 键表示操作结束。例如，修改仿真模式后要求按[MON]键，表示设置结束。

3.2.3 [Trace/Mode]键 [跟踪执行键]

上档键[Trace]

在调试程序时先按[MON]键，最左侧一只 LED 数码管出现“P”字符，这时[Trace]档起作用，每按一次[Trace/Mode]便执行一条用户编写的单片机指令，如果遇到“CALL”这类调用命令时，跟踪到调用内部。执行一条指令，可逐条检查用户程序的执行情况。

下档键[Mode]

在调试程序时按[MON]键，直至最左侧一只 LED 数码管出现“’”符号，这时下档[Mode]起作用。[Mode]的作用是设置仿真器的工作模式，仿真器共有 4 种模式

模式名称	地址位显示的字符标志	数据位显示的字符标志	模式作用对象	
			程序存储器	数据存储器
模式 1	PI	EI	仿真器	仿真器
模式 2	PI	EE	仿真器	目标板
模式 3	PE	EI	目标板	仿真器
模式 4	PE	EE	目标板	目标板

应该说明的是，这 4 种模式可在计算机上设置，也可在仿真器上设置。

仿真器模式设置方法：

1. 按[MON]键，使最左边数码管上出现“’”符号。
2. [Trace/Mode]键，数码管出现上表所示的 4 种模式中的一种，例如：“PI EE”，然后，按[Last]或[Next]键，这 4 种模式的数码显示标志轮流出现。
3. 出现所要求的模式时，按[MON]键，模式设置结束。

3.2.4 [Step/Move]键 [单步执行键]

它有上下两档，作用分别是：

[Step][单步执行键]

用户调试程序时每按一次[Step]键,执行一条命令。但是,当执行到调用语句时,按一次[Step]键将执行调用所含的所有语句，[Step]与[Trace]不同的是，[Trace]遇到调用语句将进入调用语句内部仍然一步一步地执行。

[Move][移动数据键]

此键用于将程序存储器中的某一段程序，移到另外一处，或者将外部数据存储器中的某一段数据移到另一处。

移动程序段的操作步骤如下：

1．按[MON]键，使最左侧数码管显示下档标志“P”。
2．按[Move]键，进入程序块移动状态，此时 6 位数据管显示为“0000　F”。
3．输入欲移动程序段的起始地址，例如：“1111”。
4．按[Move]键，此时 6 位数码管显示为“0000　t”。
5．输入欲移动程序段的结束地址，例如：“2222”。
6．按[Move]键，此时显示“0000　0”
7．输入 4 位目标地址，例如：“3333”。
8．按[Move]键，开始移动数据，左 4 位数码管闪烁，最后显示“P”，表示移动结束。

3.2.5 [Last]键　[地址减 1 键]

它有 3 个作用：

1. 触发显示寄存器和存储器中的数据
2. 地址减 1
3. 仿真模式的触发选择。

触发显示寄存器和存储器中的数据

刚在地址位上输入了寄存器或和存储器中的地址时，数据显示数码管是暗的未发光，此时，第一次按[Last]键，数据显示数码管开始显示数据。

地址减 1 操作

当在仿真器上输入地址后，在第二次以及之后的各次按键时，每按一次[Last]键，地址值自动减 1，地址值所对应的数据同时更换。

仿真模式的触发选择

在进行仿真模式的选择时，每按一次[Last]键，将使仿真器显示出一种模式供用户选择。

3.2.6 [Next]键　[地址加 1 键]

它有 3 个作用：

1. 触发显示寄存器和存储器中的数据
2. 地址加 1
3. 仿真模式的触发选择。

触发显示寄存器和存储器中的数据

刚在地址位上输入了寄存器或和存储器中的地址时，数据显示数码管是暗的未发光，此时，第一次按[Next]键后，数据显示数码管开始显示数据。

地址加 1 操作

当在仿真器上输入地址后，在第二次以及之后的各次按键时，每按一次[Next]键，地址值自动加 1，地址值所对应的数据同时更换。

仿真模式的触发选择

在进行仿真模式的选择时，每按一次[Next]键，将使仿真器显示出一种模式供用户选择。

3.2.7 [Here]键 [断点运行键]

设置中断程序运行的地址，使程序执行到中断地址处停止执行，在中断地址处等待新的操作命令（如 GO、STEP、TRACE 等）。设置中断点时有两种情况：

1.程序从 0000 地址处执行的断点

- 按[RST]复位，再[MON]键，数码管显示“P”。
- 输入 4 位地址。
- 按[Here]

按 Here 后，程序自动执行到断点，此时显示断点的地址和 A 寄存器中的内容。按 Here 键一是确定断点地址，同时启动了程序，并使程序执行到断点处停下来等待用户的命令。

2. 先使程序执行到某处，再设置断点

- 先用单步（Step)、跟踪（Trace)、中断等运行方式使程序执行的某处停止。
- 按“MON”键，使仿真器地址数码管显示“P”。
- 输入断点的地址值
- 按“Here”键
- 程序自动从设置前的停止处执行到所设的断点地址处。

3.2.8 [GO]键 ［全速执行键］

用仿真器调试程序时，设置好执行的初始地址后，按［GO］可自动执行程序。

3.3 键盘方式下的仿真方法

下面根据仿真时的各类需要，逐一介绍有关的操作方法。

3.3.1 将汇编源程序转换为机器码

在仿真工作之前，应把汇编源程序转换为机器码。可用人工查手册的方法逐条翻译成机器码，在翻译成机器码的同时还要为各条机器码安排地址。也可用计算机自动汇编并生成列表文件，列表文件是指同时含有源程序、机器码和机器码地址的文件。用计算机生成机器码可避免人工翻译造成的人为错误，汇编的效率也远高于人工翻译。

3.3.2 输入程序的机器码

3.3.2.1 由计算机输入程序机器码

为了节省输入机器码的时间，可先由计算机向仿真器输入机器码，然后再脱离计算机独自仿真。用计算机输入程序时先用 ：MCS51/SD1 命令启动软件，调入用户的汇编源文件，选择仿真模式，执行汇编命令，汇编结束后机器码自动装入仿真器中。然后，关断计算机和仿真器的电源，拔出 RS232 接口插头。

3.3.2.2 人工输入程序机器码

如果没有计算机时，则用人工方法输入。人工输入的方法是：

- ☛ 按［RST］键，整机复位。
- ☛ 按［MON］键，使地址数码管出现“P”字符。
- ☛ 输入 4 位地址码，此时地址位上显示的是输入的地址数，数据位上的数码管不亮。
- ☛ 先按［Next］或［Last］键，数据位上的数码管闪烁，此时，再输入 2 位数据。
- ☛ 程序输入完毕，可依次按［RST］、［MON］、4 位地址数、［Next］键，检查输入的机器码，不断地按［Next］或［Last］键，可依次逐个检查各个地址中的数据，数据形式是 16 进制。

3.3.3 执行程序

3.3.3.1 跟踪执行程序

当需要由用户通过键盘控制，逐条执行程序，以便检查单片机内部和外部电路时，可采用跟踪执行的方式。跟踪执行时如果遇到调用指令，仿真器将使程序的指针进入被调用的程序段内部。跟踪执行键［Trace ］，以跟踪方式执行程序的操作方法是：

- ☛ 按［RST］键整机复位。
- ☛ 按［MON］键使地址数码管出现字符“P”。
- ☛ 输入程序执行的起始地址，此时，4 位地址数码管显示地址值，数据数码管暗。
- ☛ 按[Trace]键，数据数码管亮，此后每按一次[Trace]键，执行一条指令。

3.3.3.2 单步执行程序

单步执行程序的作用与跟踪执行相仿，区别是单步执行遇到调用时，将自动连续地执行调用内部的所有指令，然后停在调用执行后的第一条指令上，可继续单步执行。单步执行键［Step ］的详细说明见第 3.2.4 条，以单步方式执行程序的操作方法是：

- ☛ 按［RST］键整机复位。
- ☛ 按［MON］键使地址数码管出现字符“P”。
- ☛ 输入程序执行的起始地址，此时，4 位地址数码管显示地址值，数据数码管暗。
- ☛ 按［Step ］键，数据数码管亮，此后每按一次［Step ］键，执行一条指令。

3.3.3.3 全速执行程序

当需从用户指定的程序地址处开始全速运行整个程序，或全速运行到断点处时，可采用全速执行方式。

全速执行的操作方法是：

- ☛ 按［RST］键，整机复位。

- 按［MON］键使地址数码管出现字符“P”。
- 输入程序执行的起始地址，此时，4 位地址数码管显示地址值，数据数码管暗。
- 如果需要设置断点，可按上面介绍的[Here]命令。
- 按［Exec］键，全速执行。程序运行时地址数码管数据数码管均暗。
- 按[RST] 中断运行。

3.3.3.4 检查执行结果

当运用上述的各种执行方式运行程序时，需要及时地了解程序执行的结果，而运行结果很大程度上是由单片机内部各个部分的当前值来反映的。仿真器提供了查看单片机（由仿真器模仿）内部各个部分情况的功能，检查执行结果的主要任务就是查看单片机内部的情况。

3.3.4　检查单片机内部各个寄存器、累加器、接口电路

3.3.4.1 检查和修改单片机内部寄存器数据

在调试过程中，如果需要查看当前状态下内部寄存器的情况，可按下列步骤进行：

- 查出单片机内部寄存器地址码。如 R1 地址为 01H。
- 根据内部寄存器地址码输入 2 位地址码，此时地址数码管的右 2 位显示地址数，左 2 位暗。
- 按［Next］或［Last］键，使数码位出现内部寄存器中的数值。
- 如果要查看相邻地址的寄存器的内容，可继续按［Next］或［Last］键。
- 如果要修改内部寄存器中的内容，可按数字键［0］…［F］。

3.3.4.2 检查和修改单片机内部专用寄存器（SFR）数据

在调试过程中，如果需要查看当前状态下单片机内部专用寄存器（SFR）的情况，可按下列步骤进行：

- 查出单片机内部专用寄存器（SFR）地址码。如 ACC 地址为 E0H。
- 仿真器规定专用寄存器地址要用 3 位数表示，专用寄存器的地址是 2 位，需先输入一个先导“0”，再输入 2 位专用寄存器地址码。地址输入后，此时地址数码管的右 3 位显示地址数，最左 1 位暗。
- 按［Next］或［Last］键，使数码位出现专用寄存器中的数值。
- 如果要查看相邻地址的专用寄存器的内容，可继续按［Next］或［Last］键。
- 如果要修改专用寄存器中的内容，可按数字键［0］…［F］。

注意：在查看专用寄存器时，不可按［RST］键，因为按此键后各个专用寄存器中内容将被复位。

3.3.3.5　检查和修改单片机程序存储器和外部数据存储器

- 按［MON］键，地址数码管出现“P”字样。
- 输入 4 位地址，此时 4 位地址数码管亮，2 位数据数码管暗。
- 按［Next］或[Last]键，2 位数据数码管亮，显示的即为 4 位地址单元中的机器

码值，如要修改则可按[0]…[F]键。

- ☛ 再按［Next］或[Last]键可查看相邻的地址中的机器码值。

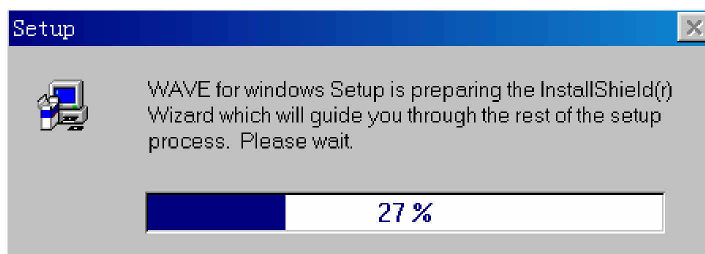
❹ 软件安装

🖫 WINDOWS 版本软件安装

❶ 将光盘放入光驱中，将在屏幕上自动显示：

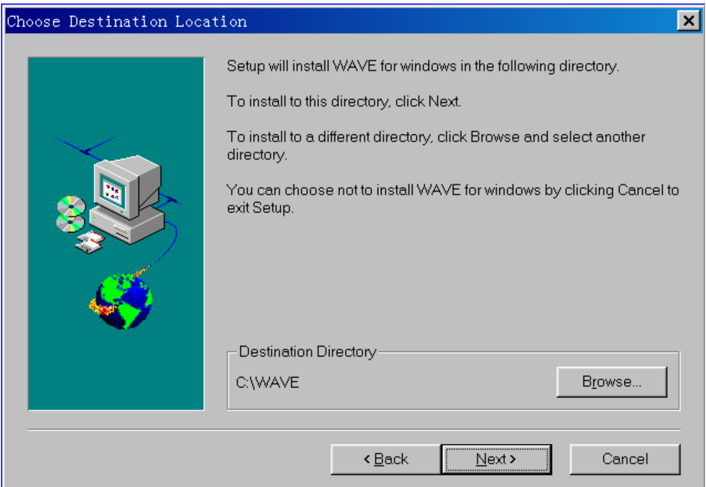


❷ 用鼠标点击[WINDOWS]行，将出现：

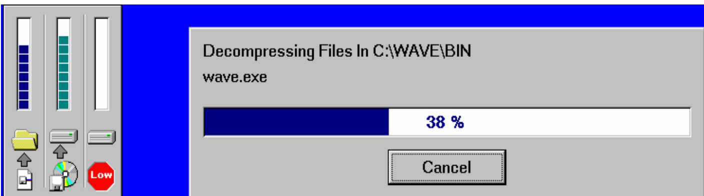


❸ 在随后的画面中都点击[NEXT]，直到出现以下画面。如要更该安装路径，请点击

[Browse]，否则，按[NEXT]继续安装。



❹ 继续点击[NEXT]，出现如下安装进度表，直至安装完毕。



❺ 还可用以下方法进入安装状态：

- 点击光盘中[ICESSOFT\E2000W]中的 SETUP

❻ **注意**:在安装新版本软件时,如果硬盘上已有老版本的软件,请完全卸载老版本软件后,再安装新软件.

❼ 在安装过程中,如果用户没有指定安装目录,安装完成后,会在 C:盘建立一个 C:\WAVE 目录(文件夹)。结构如下:

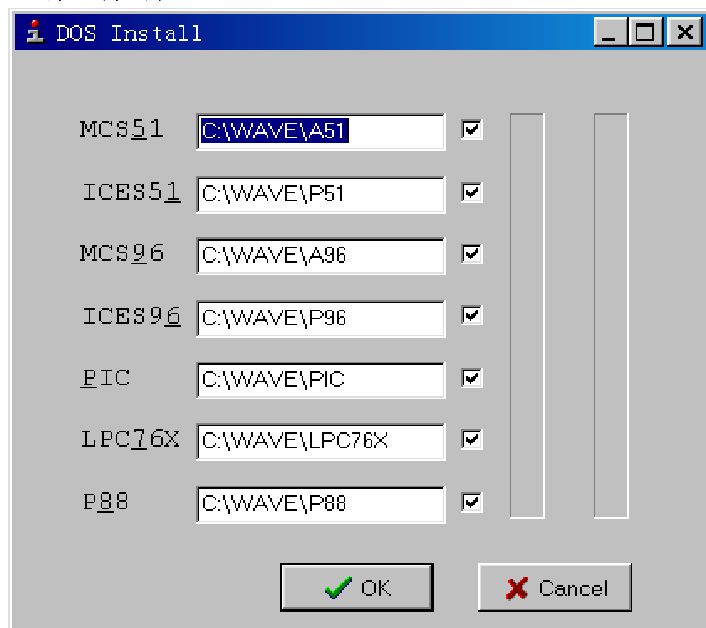
目录	内容
C:\WAVE	
├ BIN	可执行程序
├ HELP	帮助文件和使用说明
└ SAMPLES	样例和演示程序

DOS版本软件安装

❶ 将光盘放入光驱中，将在屏幕上自动显示：



❷ 用鼠标点击[DOS]行，将出现：



❸用鼠标点击[OK]，将出现安装进度表,直至完毕。于是在 C：\WAVE 目录下拥有了伟福 DOS 版调试平台。

🖫 KEIL C51 编译器安装

❶ 进入 C:盘根目录,建立 C:\COMP51 子目录(文件夹)

❷ 将光盘中[编译器]目录中的 COMP51.EXE 文件复制到 C:\COMP51 子目录(文件夹)下。

❸ 运行 COMP51.EXE 自解压文件，即可得到 KEIL C51 编译器。

❹ 如果用户将第三方编译器安装在硬盘的其它位置,请在[编译器路径]指明其位置.

🖫 程序清单安装

⓿ 程序清单安装与否，请自行考虑。

❶ 在硬盘上建立子目录，比如：D:\MCS51，D:\C51

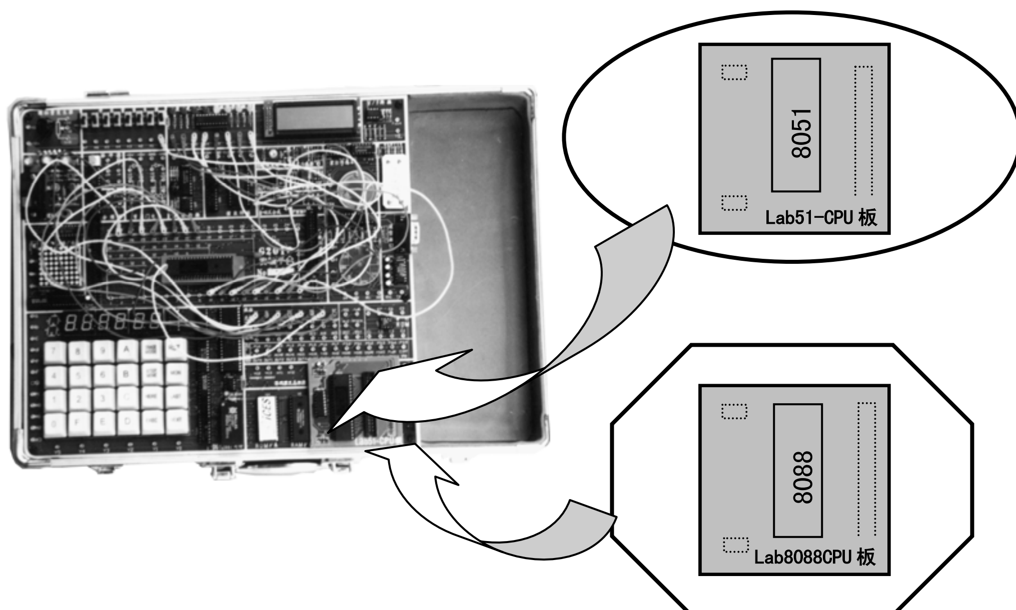
❷
- 将光盘中[G2010 程序清单\MCS51]子目录中的内容复制到 D:\MCS51\子目录(文件夹)下然后，把各文件的只读属性去了。
- 将光盘中[G2010 程序清单\C51]子目录中的内容复制到 D:\C51\子目录(文件夹)下.然后，把各文件的只读属性去了。

❸ 请不要使用长字母的目录和文件名。

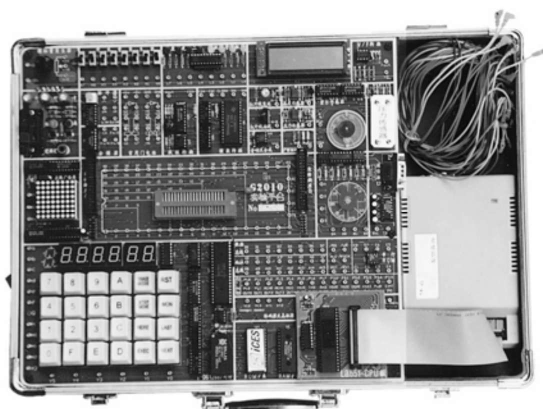
❺ 硬件安装

连接 Lab51CPU 板：

在实验平台的右下角有三个插座，是用来安装实验 Lab8051CPU 板的。

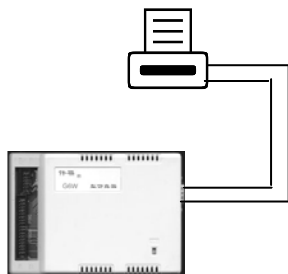


仿真器与实验平台的连接：



- 将 Lab51 板的 DC34 芯插座与 G6W 仿真器上的 DC34 插座用扁平电缆连接起来。

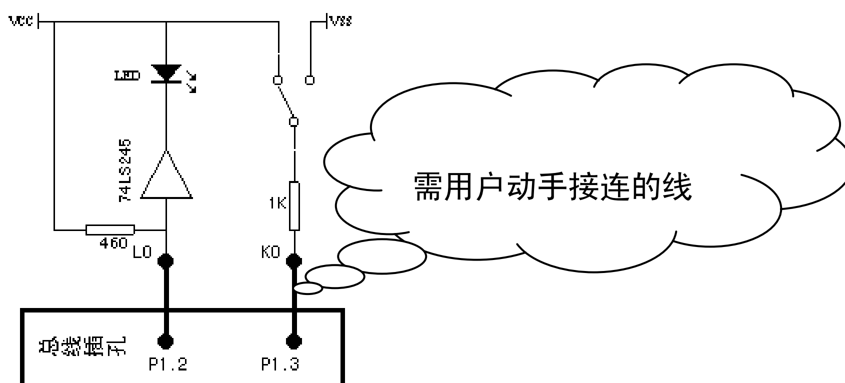
仿真器与计算机的连接：



- 用随机配带的串口通讯电缆，将仿真器与计算机连接起来，串口 1、串口 2 均可。但务必注意所选用的串口未被设置成它用，如 Modem 口、鼠标口。
- 应特别注意的是，在仿真器与计算机连接串口电缆时，两台机器必须都关断电流。否则易损坏计算机和仿真器。

实验连线：

按实验方案，用随机的实验连线插入孔后， 轻轻转动一下锁紧插头， 保证良好接触。拆线时，应先回转一下，不要硬拨， 以免损坏线路板。 不管是拆线还是插线，都应在断电的情况下进行。**实验例程中“接线方案”的粗线即为需用户动手接连的线**。



❻ 实验例程

第❶节 MCS51“验证式”实验例程

实验一　拆字程序（键盘调试）

1.实验目的：

掌握汇编语言设计方法；学习手工汇编的方法；学习键盘输入机器代码的方法；

2.实验内容：

把8000H地址上的内容拆开，高位送8001H地址的低位，低位送8002H地址的低位，8001H、8002H地址的高位清零.本程序通常在把数据送显示缓冲区时使用。

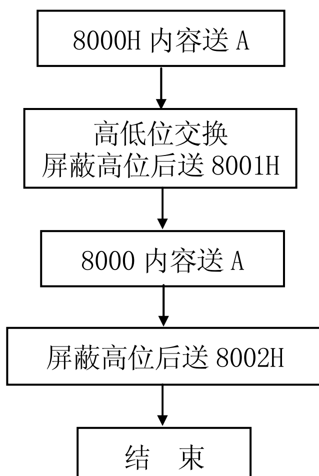
3.实验器材：

(1) G2010实验平台　1　台　　　(2) G2K仿真板　1　个

4.实验原理：

尽管手工编译由于效率太低而在实际工作中不再使用，然而对于初学者而言，手工编译对理解编译原理是必不可少的，同时对于理解一些譬如相对偏移、绝对偏移等专用名词的含义帮助甚丰。 仿真工具为了使得工具的64K的程序空间（也称仿真ROM)、64K的数据空间（也称仿真RAM）和用户的64K的程序空间（也称用户ROM)、64K的数据空间（也称用户RAM）完全分离，从而达到全资源开放（或准全资源开放）目的，分别对/PSEN、/WR、/RD信号进行了硬件处理，分别对应仿真ROM空间、仿真RAM空间、用户ROM空间、用户RAM空间，这四个空间的组合，即为四种工作模式。

5.程序框图：



6. 实验步骤：

（1）按流程图编写程序，借助 MCS51 指令速查表手工编译成机器代码。以下是通过计算机交叉汇编得到的.LST 文件清单,供参考。

```
                        ORG 0000H
0000 908000           MOV DPTR,#8000H        ;指定的字节
0003 E0               MOVX A,@DPTR
0004 F5F0             MOV B,A                ;暂存
0006 C4               SWAP A                 ;交换
0007 540F             ANL A,#0FH             ;屏敝高位
0009 A3               INC DPTR
000A F0               MOVX @DPTR,A
000B A3               INC DPTR
000C E5F0             MOV A,B
000E 540F             ANL A,#0FH             ;指定字节的内容屏敝高位
0010 F0               MOVX @DPTR,A
0011 80FE      LOOP:  SJMP LOOP
```

（2）参阅《❸G2K 仿真板（选件）》，把 G2010+实验平台与 G2K 仿真板连接好。打开电源，数码管应显示 8051，否则，请速关机。

（3）通过键盘输入程序代码:

键盘操作	LED 数码管显示					
上电	8	0	5	1		
MON	P					
0				0		
0			0	0		
0		0	0	0		

Key						
0	0	0	0	0		
NEXT	0	0	0	0	×	×
9	0	0	0	0	9	_
0	0	0	0	0	9	0
NEXT	0	0	0	1	×	×
8	0	0	0	1	8	_
0	0	0	0	1	8	0
NEXT	0	0	0	2	×	×
0	0	0	0	2	0	_
0	0	0	0	2	0	0
NEXT	0	0	0	3	×	×
E	0	0	0	3	E	_

按键	显示					
0	0	0	0	3	E	0
NEXT	0	0	0	4	×	×
F	0	0	0	4	F	_
5	0	0	0	4	F	5

┆ 按以上方法输入完程序代码 ┆

（4）外部数据 XDATA 查看及修改

按键	显示					
MON	P					
MON	’					
8				8		
0			8	0		
0		8	0	0		
0	8	0	0	0		
NEXT	8	0	0	0	X	X
6	8	0	0	0	6	_

B	8	0	0	0	6	B
NEXT	8	0	0	1	X	X

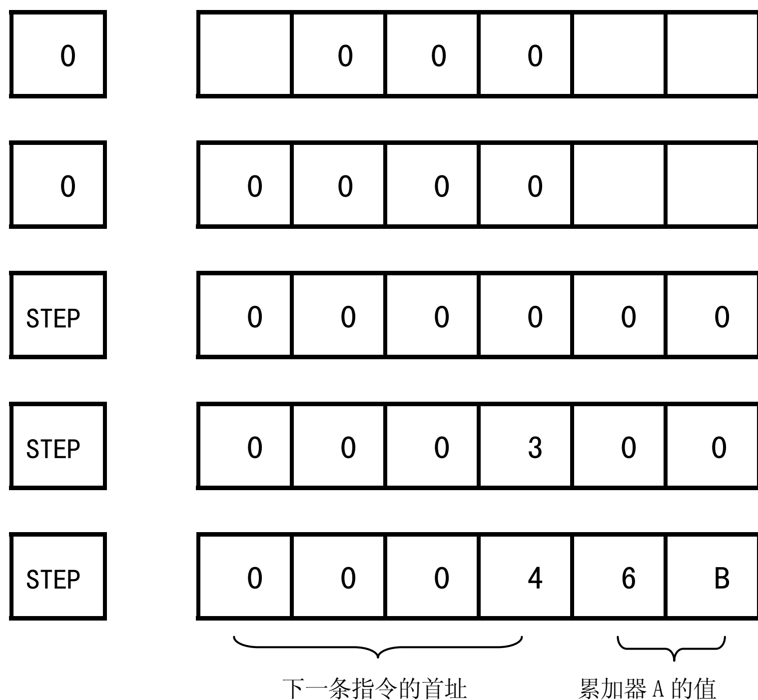
（5）设置工作模式

MON	P					
MON	’					
MODE	P	I			E	I
LAST	P	E			E	I
LAST	P	I			E	E

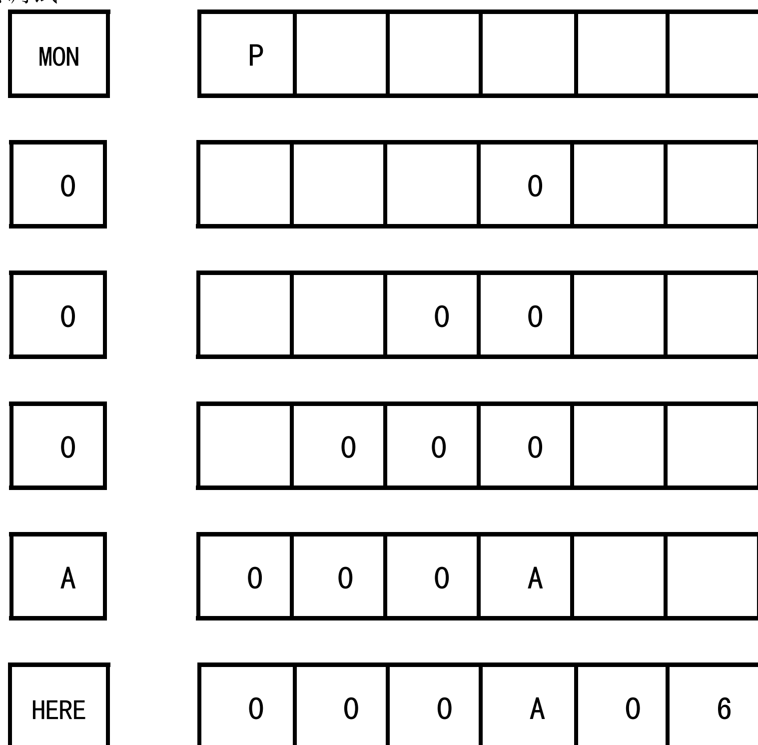
“PI　EI”为工作模式 1，即程序和数据存储器均使用仿真器的；“PI EE”为工作模式 2，即程序空间使用 G2K 仿真板、数据空间使用用户板，此时，则使用实验平台的 U2 扩展 RAM 6264，其地址空间为 8000H-9FFFH。为了更深刻的了解不同工作模式时的不同的作用空间，可把 G2010+上的 EPROM27C256（即所谓的用户 ROM）和 RAM6264（即所谓的用户 RAM）拔下，观察不同工作模式下这两片芯片的插上和拔下所得到的值有什么不同。

（6）单步调试

MON	P					
0				0		
0			0	0		


0
0 0 0
0
0 0 0 0
STEP
0 0 0 0 0 0
STEP
0 0 0 3 0 0
STEP
0 0 0 4 6 B
下一条指令的首址
累加器 A 的值

（7）断点调试


MON
P
0
0
0
0 0
0
0 0 0
A
0 0 0 A
HERE
0 0 0 A 0 6

（8）特殊功能寄存器查看

Key						
MON	P					
0				0		
E			0	E		
0		0	E	0		
NEXT		0	E	0	0	6

7.软件清单 （MSC51\A01.ASM）

实验二　拼字程序（键盘调试）

1. 实验目的：

（1）进一步掌握汇编语言设计；　　（2）学习键盘方式下的各种调试方法；

2. 实验内容：

把 8000H、8001H 两个字节的低位分别送入 8002H 的高位和低位。本程序一般用于把显示缓冲区数据取出拼装成一个字节。

3. 实验器材：

(1) G2010 实验平台　1　台　　(2) G2K 仿真板　1　个

4. 实验原理：

尽管键盘调试的方式现在已不常使用，但对于初学者而言，键盘方式下的实验对理解单片机的指令系统和空间概念还是相当有用的。比如：MOV，MOVX 指令；MOV R0，A 和 MOV @R0,A;MOVX @R0,A 和 MOVX @DPTR，A 等。

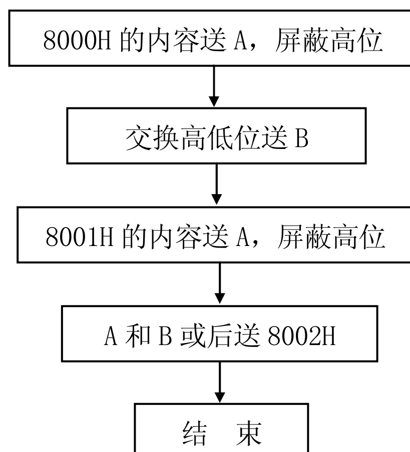
5. 实验步骤：

（1）编写程序并编译成机器代码，通过键盘把机器代码输入到实验仪中。

(2）设定仿真器的工作模式为模式 1，即程序和数据存储器均使用仿真器的。如果在模式 2 做该实验，则使用实验平台的扩展 RAM 6264，其地址空间为 8000H-9FFFH。仿真模式的设定方法：按“MON”键，使得数码管显示“ ’ ”，再按“MODE”键，使用“NEXT”、“LAST”键使得数码管显示“PI EI”（工作模式 1）或“PI EE”（工作模式 2）。

（3）用断点或单步方式运行程序，检查 8000H-8002H，A，B 中内容变化情况。查看内部 RAM、外部 RAM、断点运行方式、单步操作等请参阅有关章节。

6. 程序框图：



7. 思考问题：

修改 8000H、8001H 内容重复上述实验

8. 软件清单：（MCS51\A02.ASM）

实验三　数据区传送子程序（DOS 平台调试方法）

1.实验目的:

学习 DOS 平台下的编辑、编译、排错、调试方法。学习修改和观察变量的方法;

2.实验内容:

把外部扩展 RAM(6264)的 8000H-807FH 中的内容传送到 8080H 开始的空间中去。R2.R3 存放源 RAM 区首址，R6.R7 存放需传送的字节数，R4.R5 存放目的 RAM 区首址。

3.实验器材:

（1） G6W 仿真器　1　台　　　（2）计算机　1 台　　　（3）G2010+实验平台　1　台

4.实验步骤:

（1）参阅 Page40《❺硬件安装》把实验平台、仿真器与 PC 机串行口连起来，打开电源。



图 A

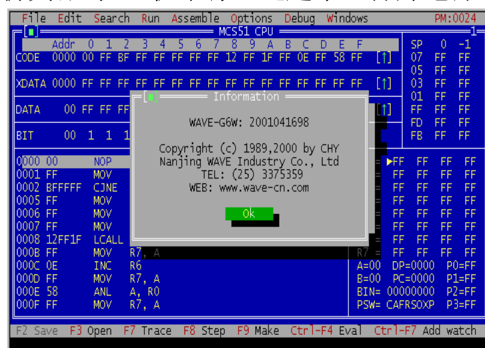


图 B

（2）在相应的安装目录中打入 MCS51/S1 如显示图 A，则需检查电源、通讯口、连接状态。有关“MCS51/S1”的含义请看下页黑框中的注释。如出现上图 B，按“Enter”键，如出现下图 C，则需进行第 3 步：初始化设置；如出现下图 D，则可进入第 4 步：建立文件。

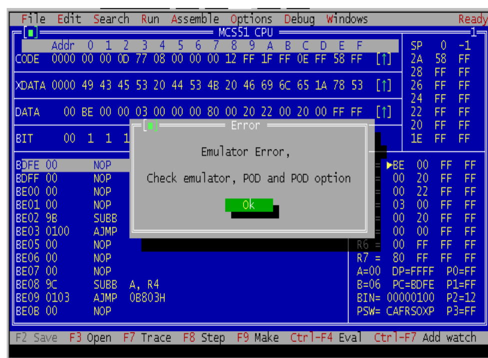


图 C

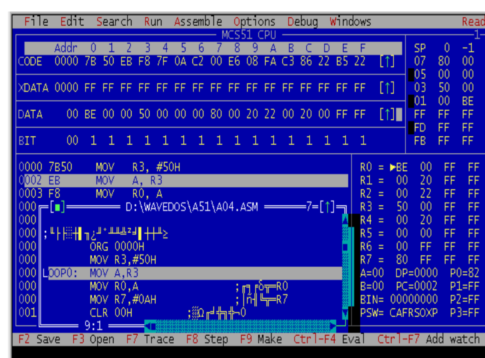


图 D

注：选项含义　/S：使用硬件仿真器;

/1：使用串口 1

/2：使用串口 2;

/I，使用 INST 区分程序/数据存储器，仅用于 MCS96 系列

/D：降低通信速率

通常情况下，你不用选择波特率，软件将自动地选择最合适的波特率进行通信，只有在使用过程中，经常出现通信出错时，才需人工调正

例：C:>MCS51;　　使用软件模拟，不需硬件仿真器

C:>MC51/S2;　　使用仿真器，串口 2 通信

（3）初试化设置：打开 Options 菜单，选择 ICE Options，在弹出的 POD Options 菜单中，POD Selet 选择为 POD51，POD552，下图 E，再点击 POD Setting，在弹出的窗口中选择 CODE 为 Emulator，Auxiliary 为 Targer，下图 F。

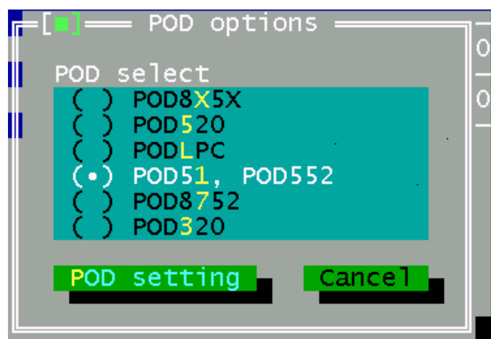


图 E

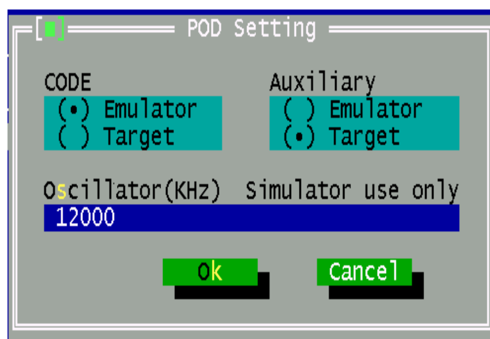


图 F

（4） 打开[File|Open]，在弹出的窗口 Name 栏输入后缀为.ASM 的文件名,再点击“Open”键即进入编辑状态。如下图 G，输入以下程序：

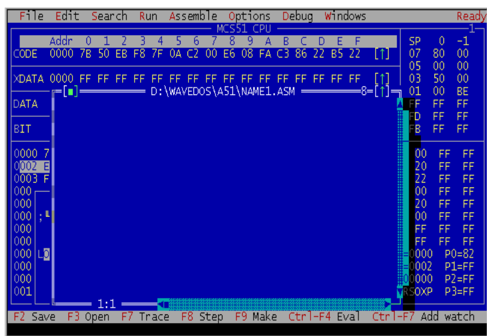


图 G

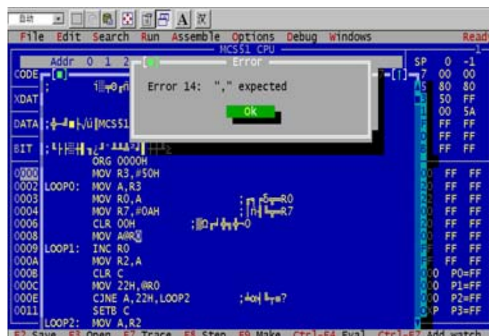


图 H

```
        ORG  0000H
LOOP0:  MOV  DPL,R3
        MOV  DPH,R2              ;建立源程序首地址
        MOVX  A,@DPTR            ;取数
        MOV  DPL,R5
        MOV  DPH,R4              ;目标首地址
        MOVX  @DPTR,A            ;传送
        CJNE  R3,#0FFH,LOOA0
        INC  R2
```

```
LOOA0:    INC  R3                          ;源地址加 1
          CJNE  R5,#0FFH,LOOP1
          INC  R4
LOOP1:    INC  R5                          ;目地址加 1
          CJNE  R7,#00H,LOOP2
          CJNE  R6,#00H,LOOP3
LOOP4:    SJMP  LOOP4
LOOP2:    DEC  R7                          ;字节数减 1
          SJMP  LOOP0
LOOP3:    DEC  R7
          DEC  R6
          SJMP  LOOP0                      ;未完继续
```

（5）按 F9 键，自动存盘并进行编译，若有语法错误，则光标停出错的语句上，在弹出的信息窗口中提示是什么类型的错误，如上图 H。若没语法错误，则自动下载目标代码至仿真器中，进入调试状态。

（6）在 R2.R3 中输入源首址< 例如 8000H>，R4.R5 中输入目地址<例如 8080H>，R6.R7 中输入字节数<例如 007FH>。输入方法：使用 Alt+I 键打开[WINDOWS|DATA] 窗，在 02H，03H 地址上输入 80H，00H；在 04H，05H 地址上输入 80H，80H；在 06H，07H 地址上输入 00H，7FH，如下图 I。使用 Alt+N 键，打开[WINDOWS|XDATA]窗，使用“↑”“↓”键或 Alt+G，使得从 8000H 开始显示数据，在此窗口 8000H 开始的地址上中输入一组任意数据后，如图 M。

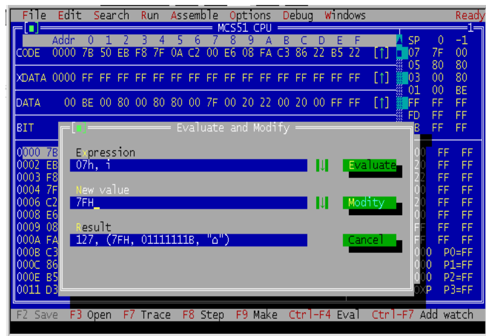


图 I

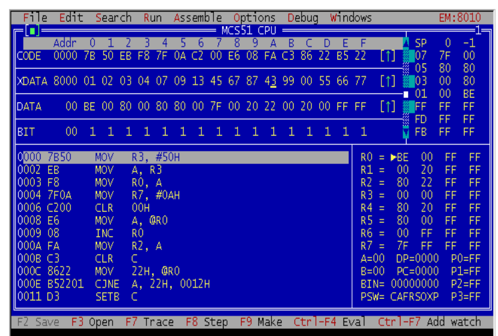


图 M

（7）单步执行程序

按 F7 一条一条的执行下去，注意 XDATA 窗口中 8080H 开始的地址上内容以及寄存器窗口中 A，DPTR，R0 等的变化情况，此程序中的加亮条表示下一条要执行的指令。

（8）按 CTRL+F9 键实时执行程序

（9）中断执行程序

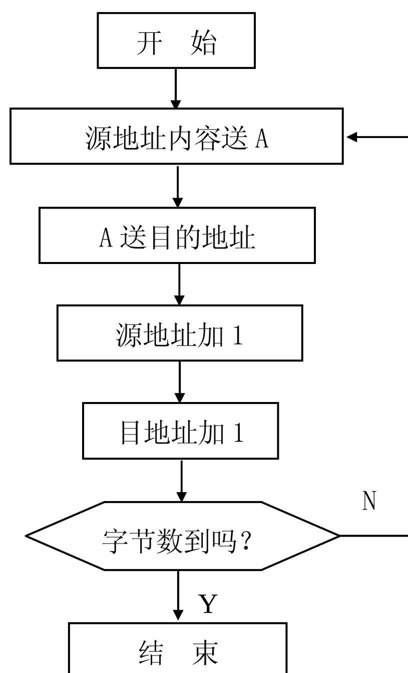
若是硬断点仿真打入 CTRL+C 即可返回监控，若是软件断点仿真器，则按仿真器上的 RESET 键进行硬件复位。此时，请观察 8080H-80FFH 之间的内容是否与 8000H-807FH 之间的内容一致。

（10）断点执行程序：要使程序执行到某条指令处暂停，如希望程序执行到 LOOA0 处暂停，可按如下操作：将光标移至 LOOA0 所在行，再按 F4。另一方法是：光标移至 LOOA0 所在行，再

按 Ctrl+F8 键，设置断点，再按 Ctrl+F9 键运行程序，将在 LOOPA0 行停止运行。

（11）软件复位 CPU :按 CTRL+F2

5.程序框图：



6.软件清单：（MCS51\A03.ASM）

实验四　数据排序实验（WINDOWS 平台调试方法）

1. 实验目的：

（1）学习 WINDOWS 平台下的编辑、编译、排错、调试方法。

（2）学习修改和观察变量的方法；综合使用单步、断点调试的方法。

2. 实验内容：

编写并调试一个排序子程序，其功能为用冒泡法将内部 RAM 中几个单字节无符号正整数，按从小到大的次序重新排列。

3. 实验器材：

(1) G6W 仿真器　1　台　　（2）计算机　1 台　　（3）G2010+实验平台　1　台

4. 实验步骤：

（1）参阅 Page40《❺硬件安装》把实验平台、仿真器与 PC 机串行口连起来，打开电源。

（2）在 PC 机上用鼠标点击“Wave”图标，进入 WINDOWS 调试环境，而后进行初始化设置。点击[仿真器\仿真器设置]出现“仿真器设置”窗，如图 A，点击此窗中[仿真器]，依次选择仿真器为 G6W，选择仿真头为 POD-51，选择 CPU 为 8032，然后点击此窗中[仿真头设置]，弹出“仿真设置”窗，如图 B，程序存储器的设置和外部数据存储器设置的组合即四种工作模式。这里设定程序存储器在仿真器上，数据存储器在用户板上。



图 A

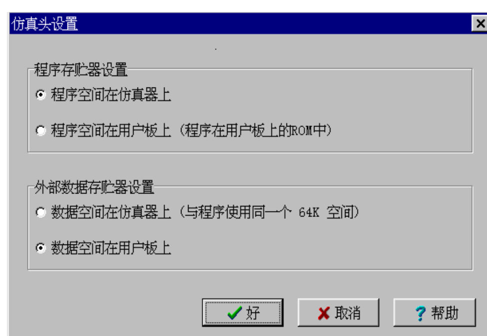


图 B

（3）去除[使用伟福软件模拟器]之前“√”，而后点击[通信设置]选择串口后，点击[好]键，对仿真器初始化，如连接成功，则出现如图 C；如连接不成功，则出现如图 D。

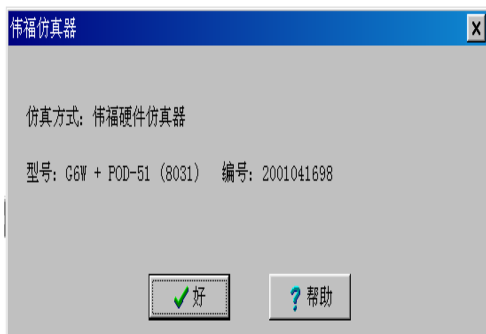


图 C

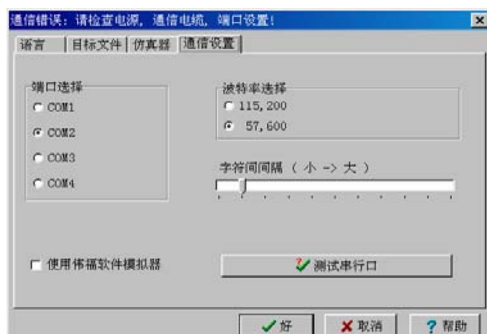


图 D

（4）在连机成功的基础才可进行下一步操作。用鼠标点一下[文件\打开文件]键，输入文

件名 NAME1.ASM，创造新文件，进入编辑窗口，在该窗口中输入程序，点击保存键存盘。

```
          ORG  0000H
          MOV  R3,#50H
LOOP0:    MOV  A,R3
          MOV  R0,A            ;指针送 R0
          MOV  R7#0AH          ;长度送 R7
          CLR  00H             ;标志位为 0
          MOV  A,@R0
LOOP1:    INC  R0
          MOV  R2,A
          CLR  C
          MOV  22H,@R0
          CJNE  A,22H,LOOP2    ;相等吗?
          SETB  C
LOOP2:    MOV  A,R2
          JC  LOOP3            ;小于或等于不交换
          SETB  00H
          XCH  A,@R0
          DEC  R0
          XCH  A,@R0
          INC  R0              ;大于交换位置
LOOP3:    MOV  A,@R0
          DJNZ  R7,LOOP1
          JB  00H,LOOP0        ;一次循环中有交换继续
LOOP:     SJMP  LOOP           ;无交换退出
```

（5）用鼠标点[项目\全部编译]窗，对当前窗口的源文件进行编辑。如果编译有错，信息窗口将自动弹出来，如图 E，并表明出错的状态和地方，用鼠标双击信息窗口信息条，则自动定位源文件中出错的地方，你可修改存盘后再编译，直到编译通过为止。



图 E

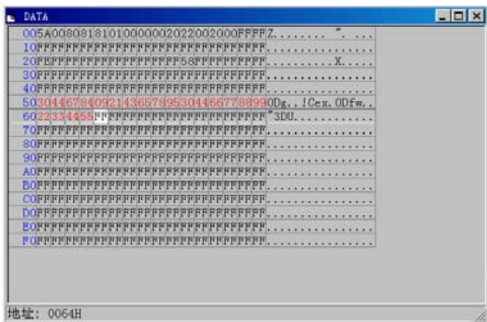


图 F

（6）打开[窗口|数据窗口|DATA]，在弹出的 DATA 窗口中 50H 开始的地址上输入不同的 10



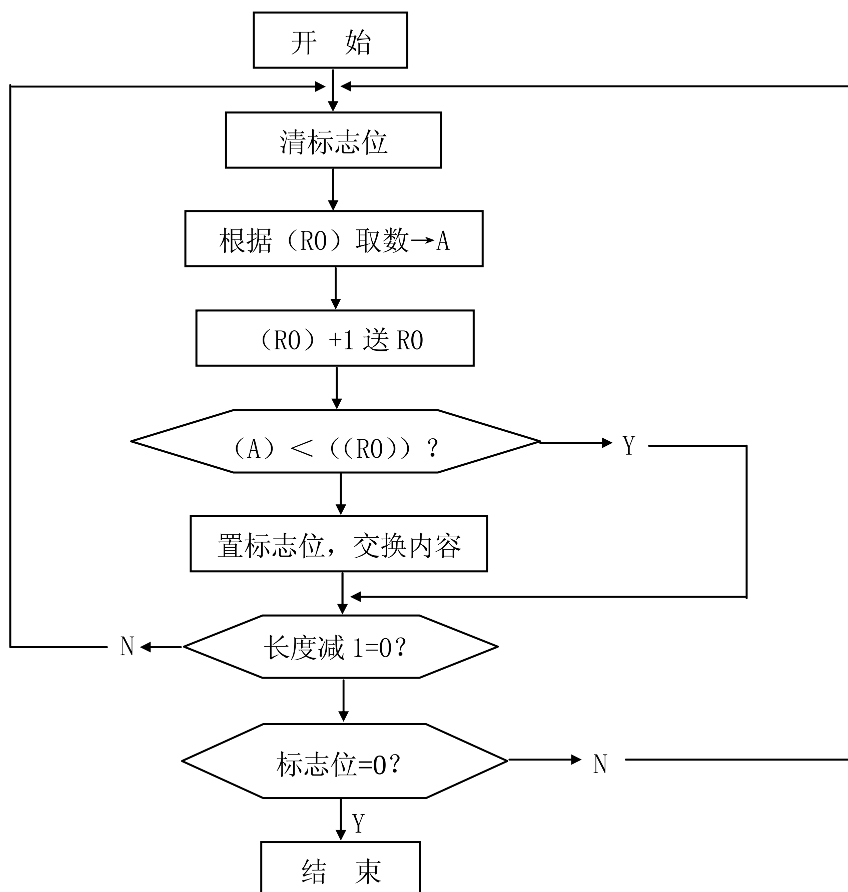
个数，如下图 G。

```
DATA
00 5A008081810100000020220020 00FFFF Z........ ".  ...
10 FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF ................
20 FEFFFFFFFFFFFFFFFFFF58FFFFFFFFFF ..........X.....
30 FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF ................
40 FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF ................
50 30446784092143657895304466778899 0Dg..!Cex.0Dfw..
60 2233445 5FFFFFFFFFFFFFFFFFFFFFFFF "3DU............
70 FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF ................
80 FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF ................
90 FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF ................
A0 FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF ................
B0 FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF ................
C0 FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF ................
D0 FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF ................
E0 FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF ................
F0 FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF ................
地址：0064H
```

图 G　　　　　　　　　　　　　　　　　　　图 F

（7）把打开的窗口进行并排，如图 F，以单步运行（F7 键），跟踪运行（F8 键）运行程序，观察 DATA 窗口 50H 开始的单元内容的变化。

5. 程序框图：



6. 思考问题：

编一程序把 50H-5AH 中内容按从大到小排列。

7. 软件清单：(MCS51\A04.ASM）

实验五　清零程序（模拟调试方法）

1.实验目的：

学习 DOS 软件模拟的调试方法

2.实验内容：

把外部 RAM（XDATA）的 2000-20FFH RAM 空间置零

3.实验器材：

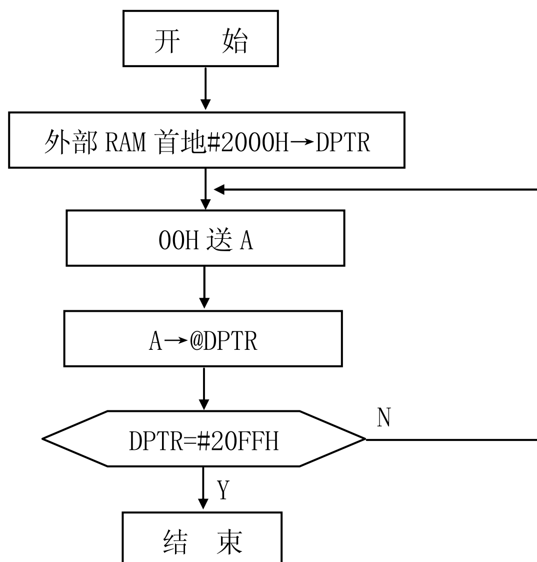
计算机　1　台

4.准备工作：

时至今日，软件模拟功能已被作为评判仿真器性能高低的标准之一。作为初学者来说可在无仿真器的情况下，即开始实验。作为实验室来说，为达到一机多用之目的提供了条件。而对于开发者而言，不仅可在无仿真器的情况下，即开始编程调试，而且，在查找问题时，可为判断是软件问题还是硬件故障提供捷径。

伟福 WINDOWS 和 DOS 平台都提供了软件模拟功能。尤其是 DOS 版功能强劲，它可以模拟包括 MCS51 5 个中断源在内的所有 CPU 资源。以下例子以 DOS 版平台为调试环境，详细说明请参考南京伟福实业有限公司《伟福系列仿真系统使用说明书》中的有关章节。

5.程序框图：



6.实验步骤：

（1）进入安装目录，执行 MCS51.EXE 文件，进入 DOS 调试环境

（2）在打开的 Open　file 窗口中的 Name 栏输入文件名，按〈Enter〉键，即进入编辑环境，请输入以下程序：

```
            ORG 0000H
            MOV R0,#00H
            MOVX DPTR,#2000H   ;空间首地址送 DPTR
      LOOP: MOV A,#00h
```

```
        MOV @DPTR,A             ;清零
        INC DPTR                ;DPTR 加 1
        INC R0                  ;字节数加 1
        CJNE R0,#00H,LOOP       ;连续清 256 个字节
LOOP1:  SJMP LOOP1
```

此程序是把 2000-20FFH 空间清零。

(3) 按 F2 键,把程序存盘。

(4) 按 F9 键（MASK）进入编译状态。弹出一个“ Error ”窗:“Error 2 Bad Operand”，这说明程序中某语句有提示中的错误，按〈Enter〉键，则光标定位于出错处。把“MOVX”改成“MOV”后，再按 F9 键，弹出一个“Assembling”窗表示编译通过。再按〈Enter〉，则进入调试状态,在第一条语句上出现光带。

(5) 按 F8 键，可以看到 R0、DPTR 随着 F8 单步运行而变化，同时，光带亦在移动。

(6) 把光标移至 XDATA 窗口，使用“↑↓”键或 Alt+G 键，使得 XDATA 窗口从 2000H 开始显示。继续按 F8 键，观察变化。

(7) 把光标移至 LOOP1 语句上，F4 键，再来观察 XDATA 窗口。可用 F5 放大显示。

7. 思考问题：

修改程序把 4000H-5000H 中内容置 55H。

8. 软件清单：（MCS51\A05.ASM）

实验六　　定时/记数器（模拟调试方法）

1.实验目的：

学习 DOS 软件模拟的调试方法。

2.实验内容：

定时/记数器的模拟调试。

3.实验器材：

计算机　　1　　台

4.准备工作：

MCS-51 单片机有两个 16 位的定时计数器 T0 和 T1,它们都有定时和对外部事件进行计数的功能，可用于定时控制、对外部事件检测和计数等场合。计数和定时实质上都是对脉冲信号进行计数，只不过脉冲源不同而已。当工作在定时方式时，计数脉冲来自单片机的内部，即振荡器信号 12 分频后作计数脉冲，每个机器周期的时间使计数器加 1，由于计数脉冲的频率是固定的（即每个脉冲为 1 个机器周期的时间），故可通过设定计数值来实现定时功能；当工作在计数方式时，计数脉冲来自单片机的引脚，每当引脚上出现一个脉冲时，计数器加 1，从而实现计数功能。可以通过编程来指定定时/计数器的功能，以及它的工作方式。

本实验中作计数器用，利用计算机键盘的 F5、F6 模拟产生脉冲信号。在实际工程中，可以用来判断记数器系统是软件问题还是硬件故障。

5.实验步骤：

（1）输入以下程序：　该程序为记数器 0 对外部输入的脉冲计数。

```
ORG 0000h
 MOV TMOD,#05h
 MOV TH0,#00H
 MOV TL0,#00H
 SETB TR0
 SETB EA
 SJMP $
```

（2）点击[Windows\Watch]，打开“Watch　and　Trap”窗，再按 Ctrl+F7 键，打开“Add Watch”窗，输入：TH0，I ，按“Enter”键，TH0 即成为观察项。同样的方法使 TL0 成为观察项。

（3）按 Ctrl+F9 键，实时运行程序。

（4）按 F5（T0 脚置低电平）再按 F6（T0 脚置高电平）产生脉冲，重复按 F5、F6 键，记下重复次数。

（5）按 Ctrl+C 键中止程序运行，观察 TL0 的值是否与输入脉冲数一样。

6.软件清单（MCS51\A06.ASM）

实验七　　中断系统（模拟调试方法）

1.实验目的：

学习 DOS 软件模拟的调试方法。

2.实验内容：

外部中断 1 的模拟调试方法。

3.实验器材：

计算机　1　台

4.实验步骤：

该程序:使用外部中断 1，电平触发方式。

（1） 输入以下程序：

```
            ORG 0000H
            SJMP MAIN
            ORG 0013H
            LJMP INTER1
MAIN:       ORL IE,#84H
            ORL IP,#04H
            SJMP $
INTER1:     NOP
            NOP
            RETI
```

（2）光标移至 INTER1 处，按 Ctrl+F8 设断点，再按 Ctrl+F9 键，实时运行程序。

（3）按 F4（INT1 脚置低电平），响应断点，表明中断被响应。

5.软件清单：(MCS51\A07.ASM)

实验八　串行口(模拟调试方法)

1.实验目的：

学习 DOS 软件模拟的调试方法。

2.实验内容：

串行口的模拟调试方法。

3.实验器材：

计算机　1　台

4.实验步骤：

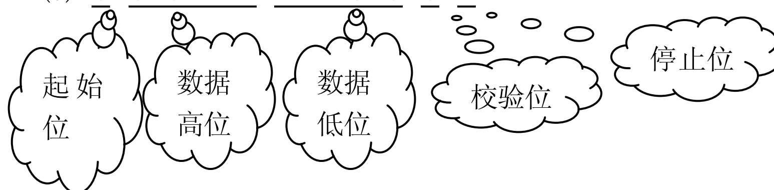
⑴ 输入以下程序

```
        ORG 0000H
        MOV TMOD,#20H
        MOV TH1,#0F3H
        MOV TL1,#0F3H
        MOV SCON,#50H
        MOV PCON,#80H
        SETB TR1
        SETB EX0
        SETB EA
DD:     JBC RI, DD1
        SJMP DD
DD1:    MOV A,SBUF
        NOP
        SJMP $
```

（2）该程序为串行通信接收方式。按 F9 键编译通过后，把光标移至 SJMP　$处，按 F4 键，程序 running。

⑶ 按 F9 键，开出 Serice　Port 窗，输入被接收数据，例：00011110001B，按〈Enter〉后，响应断点，光带同时移至 SJMP　$处。此时观察 A=3CH。

⑶ 0 0 0 1 1 1 1 0 0 0 1 B



5.软件清单：(DEMO\A08.ASM)

实验九　P1 口输入输出实验

1.实验目的：

(1)掌握 P1 口作为 IO 口时的使用方法。

(2)理解读引脚和读锁存器的区别。

2.实验内容：

P1.3 脚的状态来控制 P1.2 的 LED 亮灭。

3.实验器材：

（1）G2010+实验平台　1　　台

（2）G6W 仿真器　　1　　台

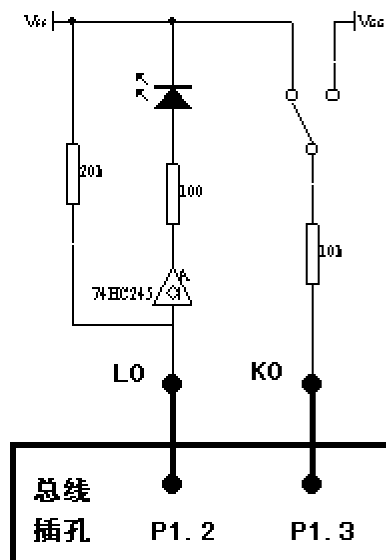
（3）连线　　　　若干　根

（4）计算机　　　　1　　台

4.实验原理:

由 8051 组成的单片机系统通常情况下，P0 口分时复用作为地址、数据总线，P2 口提供 A15-A8 即高 8 位地址，P3 口用作第二功能，只有 P1 口通常用作 I/O 口。P1 口是 8 位准双向口，它的每一位都可独立地定义为输入或输出，因此既可作为 8 位的并行 I/O 口，也可作为 8 位的输入输出端。当工作在输入方式时，对应位的锁存器必须先置 1,才能正确地读到引脚上的信号,否则,执行读引脚指令时,若对应位的锁存器的值为 0,读的结果永远为 0。每个 I/O 端口都有两种读入，即读锁存器和读引脚，读引脚指令一般都是以 I/O 端口为源操作数的指令，如 MOV　C，P1.3，而读锁存器指令一般为“读-修改-写”指令，如 ANL P1.3，C 指令，请同学们在实验中体会。图示中，P1.2 作为输出口，P1.3 作为输入口。

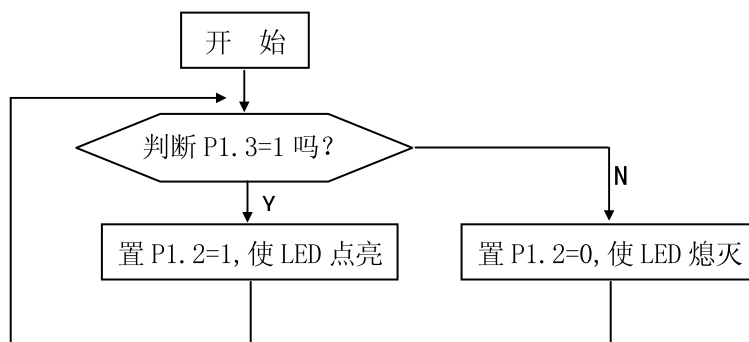
5.接线图案：



“总线插孔”框中 P1.2 孔连“发光二极管组”的 L0 孔，

P1.3 孔连“开关量发生器”的 K0 孔。

6.程序框图：



7.实验步骤：

（1）编写程序实现当 P1.3 为低电平时，发光管高；P1.3 为高电平时，发光管灭。

（2）修改程序在执行读 P1.3 之前，先执行 CLR P1.3，观察结果是否正确，分析在第二种情况下程序为什么不能正确执行，理解读引脚和读锁存器区别。

8.软件清单：（MCS51\A09.ASM）

实验十　P3.0 口输入.P1 口输出

1.实验目的：

掌握 P3 口、P1 口简单使用。

2.实验内容：

P3.0 口输入一脉冲，控制 P1 口按 16 进制加一方式点亮发光二极管。

3.实验器材：

（1）G2010+实验平台　1　　台
（2）G6W 仿真器　　　1　　台
（3）连线　　　　　若干　根
（4）计算机　　　　　1　　台

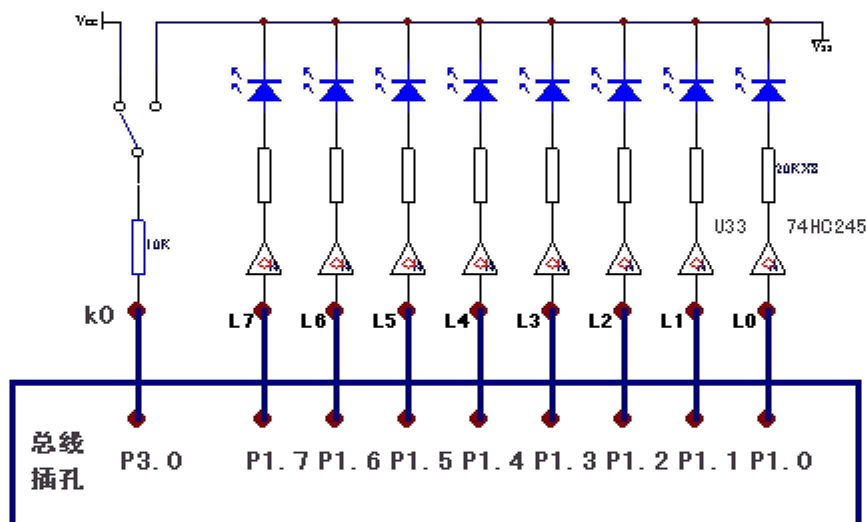
4.实验原理:

由 8051 组成的单片机系统通常情况下，P0 口分时复用作为地址、数据总线，P2 口提供 A15-A8 即高 8 位地址，P3 口用作第二功能，只有 P1 口通常用作 I/O 口。P1 口是 8 位准双向口，它的每一位都可独立地定义为输入或输出，因此既可作为 8 位的并行 I/O 口，也可作为 8 位的输入输出端。当工作在输入方式时，对应位的锁存器必须先置 1,才能正确地读到引脚上的信号,否则,执行读引脚指令时,若对应位的锁存器的值为 0,读的结果永远为 0。每个 I/O 端口都有两种读入，即读锁存器和读引脚，读引脚指令一般都是以 I/O 端口为源操作数的指令，如 MOV　C，P1.3，而读锁存器指令一般为“读-修改-写”指令，如 ANL　P1.3，C 指令。

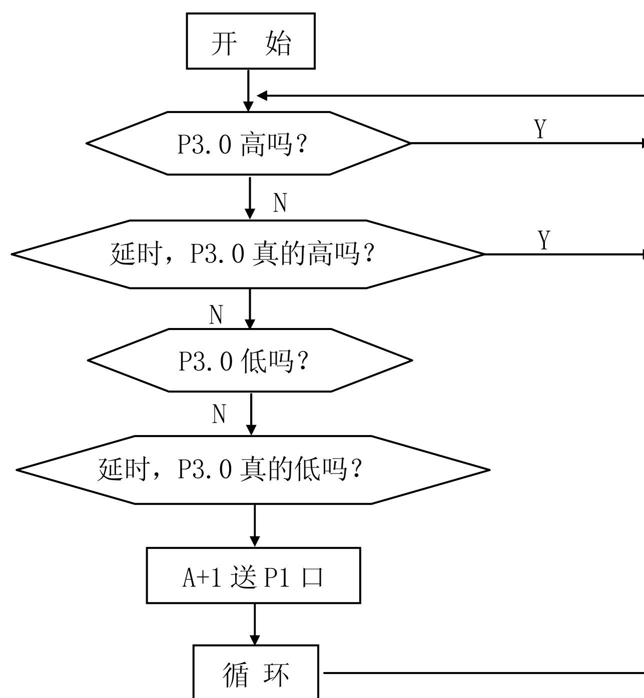
5.实验步骤：

P3.0 孔用连线连至 K0，P1.0-P1.7 孔连至 L0-L7，K0 拨动一次，L0-L7 发光二极管按 16 进制方式加一闪亮。

6. 接线图案：



7.程序框图：


开　始
P3.0 高吗？
Y
N
延时，P3.0 真的高吗？
Y
N
P3.0 低吗？
N
延时，P3.0 真的低吗？
A+1 送 P1 口
循 环

实验十一　八段数码管显示

1.实验目的:

(1)了解数码管动态显示的原理。

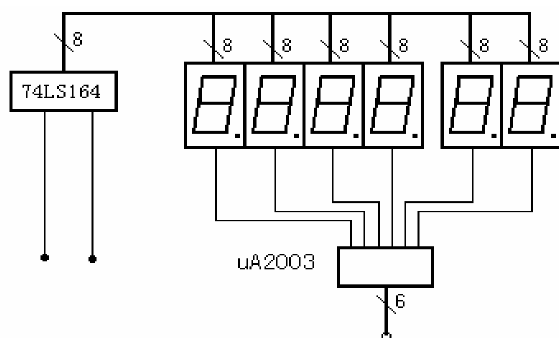
(2)了解 74LS164 扩展端口的方法。

2.实验要求:

利用实验仪提供的显示电路,动态显示一行数据.

3.实验线路:

这里只是显示草图，详细原理参见第一章的 1.1.15 “8155 键显模块”



4.实验器材：

（1)G2010+实验平台　1　台　　（2)G6W 仿真器　1　台　　（3)计算机　1　台

5.实验说明:

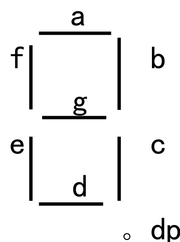
(1)本实验仪提供了 8 段码 LED 显示电路，学生只要按地址输出相应数据，就可以实现对显示器的控制。显示共有 6 位，用动态方式显示。8 位段码是由 8155 的 PB0、PB1 经 74LS164“串转并”后输出得到。6 位位码由 8155 的 PA0 口输出，经 uA2003 反向驱动后，选择相应显示位。

74LS164 是串行输入并行输出转换电路，串行输入的数据位由 8155 的 PB0 控制，时钟位由 8155 的 PB1 控制输出。写程序时，只要向数据位地址输出数据，然后向时钟位地址输出一高一低两个电平就可以将数据位置到 74LS164 中，并且实现移位。向显示位选通地址输出高电平就可以点亮相应的显示位。

本实验仪中数据位输出地址为 0e102H，时钟位输出地址为 0e102H，位选通输出地址为 0e101H。

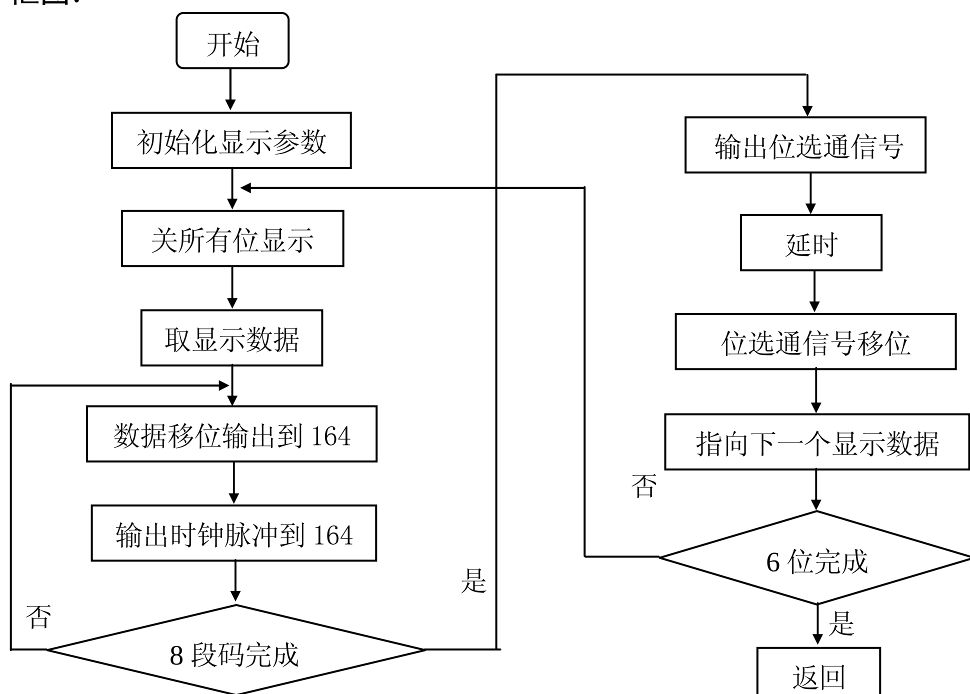
本实验涉及到了 8155 I/O/RAM 扩展芯片的工作原理以及 74LS164 器件的工作原理。

（2）七段数码管的字型代码表如下表：



显示字形	g	f	e	d	c	b	a	段码
0	0	1	1	1	1	1	1	3fh
1	0	0	0	0	1	1	0	06h
2	1	0	1	1	0	1	1	5bh
3	1	0	0	1	1	1	1	4fh
4	1	1	0	0	1	1	0	66h
5	1	1	0	1	1	0	1	6dh
6	1	1	1	1	1	0	1	7dh
7	0	0	0	0	1	1	1	07h
8	1	1	1	1	1	1	1	7fh
9	1	1	0	1	1	1	1	6fh
A	1	1	1	0	1	1	1	77h
b	1	1	1	1	1	0	0	7ch
C	0	1	1	1	0	0	1	39h
d	1	0	1	1	1	1	0	5eh
E	1	1	1	1	0	0	1	79h
F	1	1	1	0	0	0	1	71h

6．程序框图：



7.程序清单：(MCS51\A11.ASM)

实验十二　键盘扫描显示实验

1. 实验目的:

(1)掌握键盘和显示器的接口方法和编程方法。

(2)掌握键盘扫描和 LED 八段码显示器的工作原理。

2. 实验要求:

在上一个实验的基础上,利用实验仪提供的键盘扫描电路和显示电路,做一个扫描键盘和数码显示实验，把按键输入的键码在六位数码管上显示出来。

实验程序可分成三个模块。

①键输入模块：扫描键盘、读取一次键盘并将键值存入键值缓冲单元。

②显示模块：将显示单元的内容在显示器上动态显示。

③主程序：调用键输入模块和显示模块。

3. 实验器材：

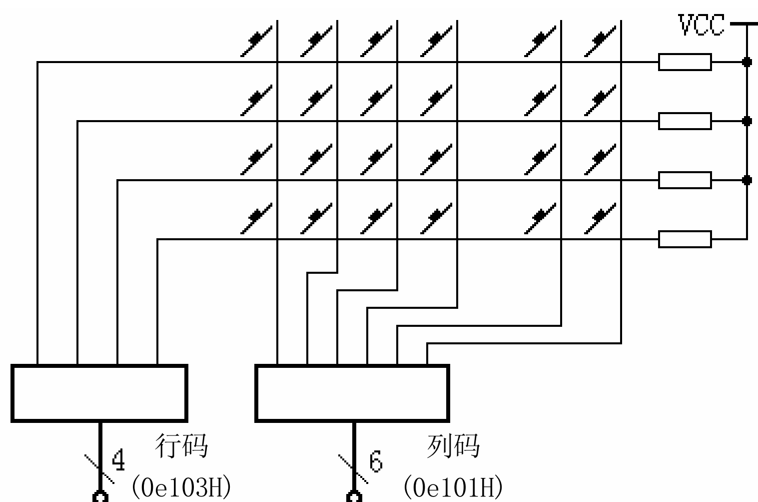
（1）G2010+实验平台　1　台

（2）G6W 仿真器　　　1　台

（3）计算机　　　　　1　台

4. 实验电路:

这里只是键盘草图，详细原理参见第一章的 1.1.15 “8155 键显模块”

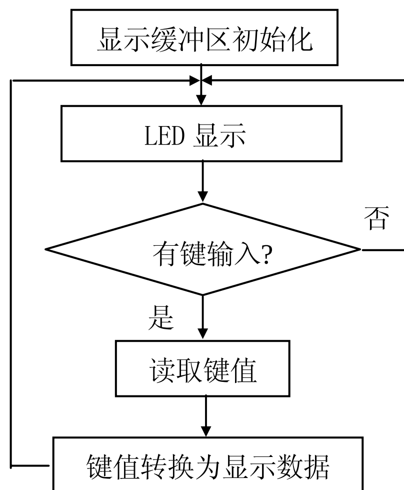


5. 实验说明:

本实验仪提供了一个 6×4 的小键盘,向列扫描码地址(0e101H)逐列输出低电平,然后从行码地址(0e103H)读回.如果有键按下,则相应行的值应为低,如果无键按下,由于上拉的作用,行码为高.这样就可以通过输出的列码和读取的行码来判断按下的是什么键. 在判断有键按下后,要有一定的延时,防止键盘抖动.

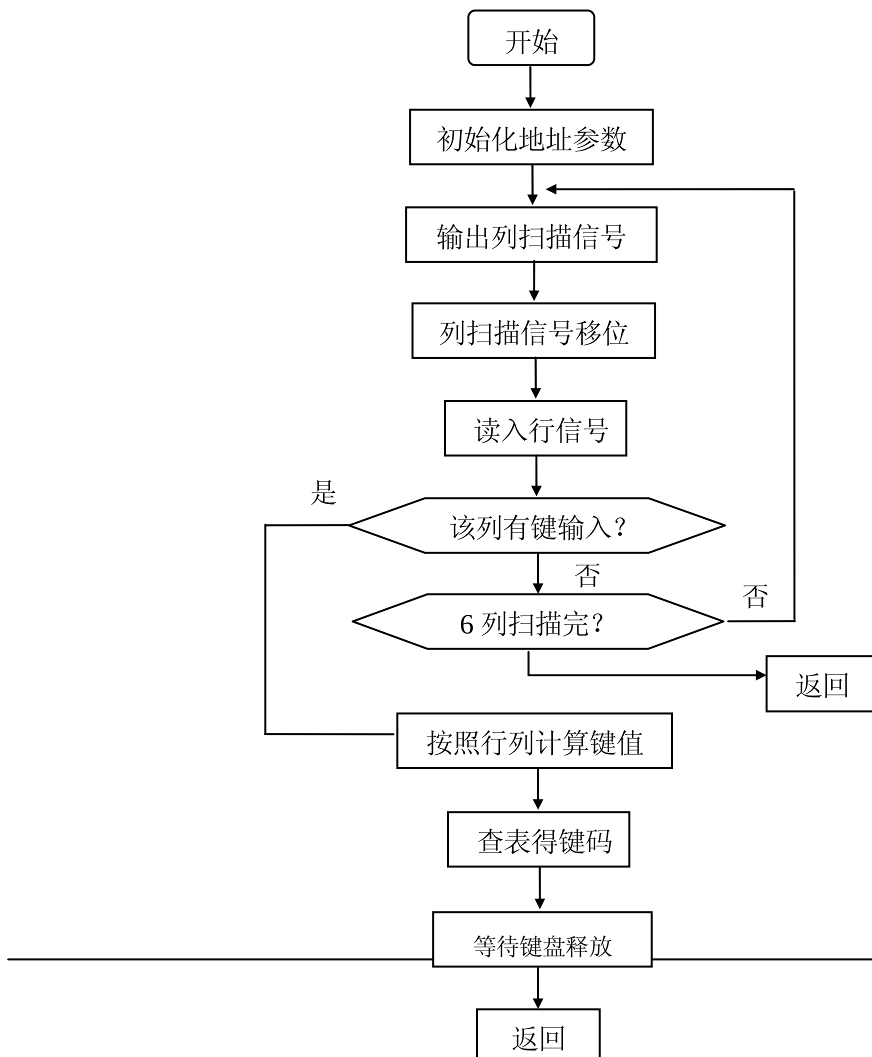
列扫描码还可以分时用作 LED 的位选通信号.

6. 实验框图：



显示缓冲区初始化
LED 显示
否
有键输入?
是
读取键值
键值转换为显示数据

主程序框图



开始
初始化地址参数
输出列扫描信号
列扫描信号移位
读入行信号
是
该列有键输入？
否
否
6 列扫描完？
返回
按照行列计算键值
查表得键码
等待键盘释放
返回

读键输入子程序框图

显示程序框图见上个实验

7.程序清单：(MCS51\A12.ASM)

实验十三　　脉冲计数（定时/计数器的记数功能实验）

1. 实验目的:

（1）熟悉 8031 定时/计数器的记数功能，

（2）掌握初始化编程方法

（3）掌握中断程序的调试方法

2. 实验内容:

定时/记数器 0 对外部输入的脉冲进行计数，并送显示器显示。

3. 实验器材:

（1）G2010+ G6W 仿真器　1　台

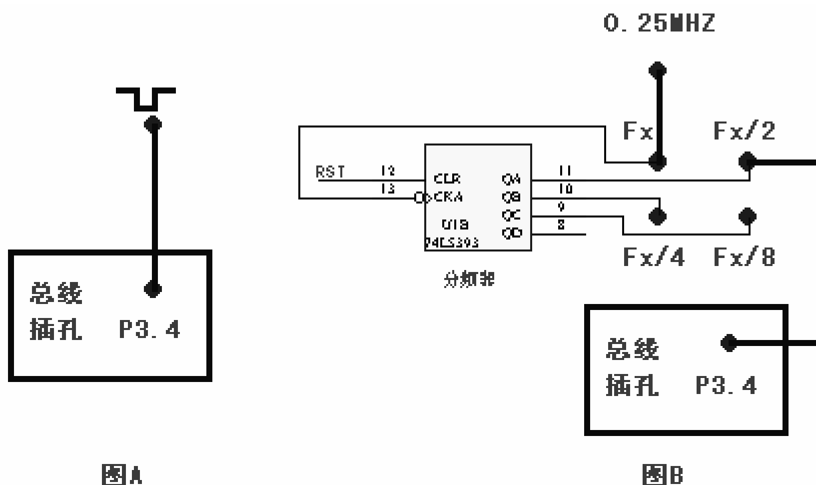
（2）连线　若干　根

（3）计算机　1　台

4. 实验原理:

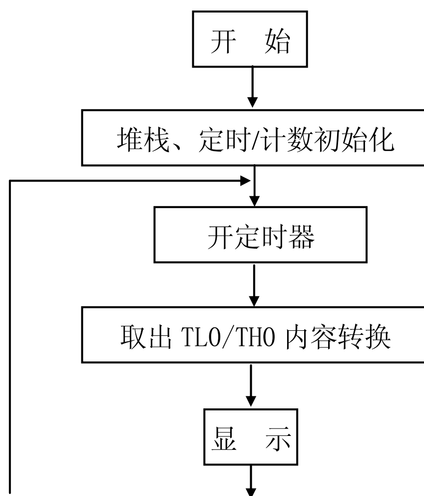
MCS-51 有两个 16 个的定时/计数器:T0 和 T1。计数和定时实质上都是对脉冲信号进行计数，只不过脉冲源不同而已.当工作在定时方式时，计数脉冲来自单片机的内部，每个机器周期使计数器加 1，由于计数脉冲的频率是固定的（即每个脉冲为 1 个机器周期的时间)，故可通过设定计数值来实现定时功能。当工作在计数方式时，计数脉冲来自单片机的引脚，每当引脚上出现一个由 1 到 0 的电平变化时，计数器的值加 1，从而实现计数功能。可以通过编程来指定时计数器的功能，以及它的工作方式。读取计数器的当前值时，应读 3 次。这样可以避免在第一次读完后，第二次读之前，由于低位溢出向高位进位时的错误。

5. 接线图案：

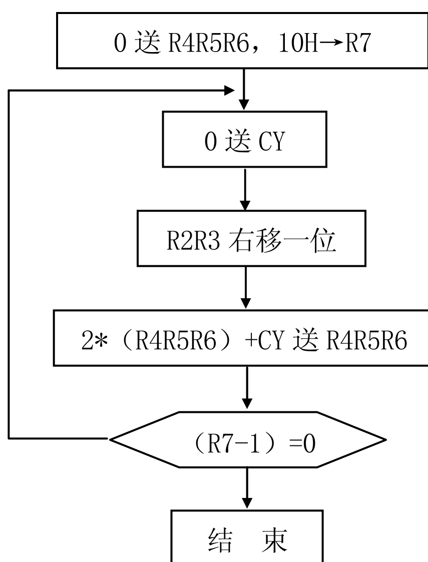


图A　　图B

6. 程序框图：



主程序流程图



二转十进制子程序

7. 实验步骤:

用连线把“总线插孔”的 P3.4 孔连“脉冲源”的“┓┏”孔 ,执行程序，按动 TR3 带锁按钮，观察数码管上计数脉冲的个数。

8. 软件清单: （MCS51\A13.ASM）

9. 思考问题:

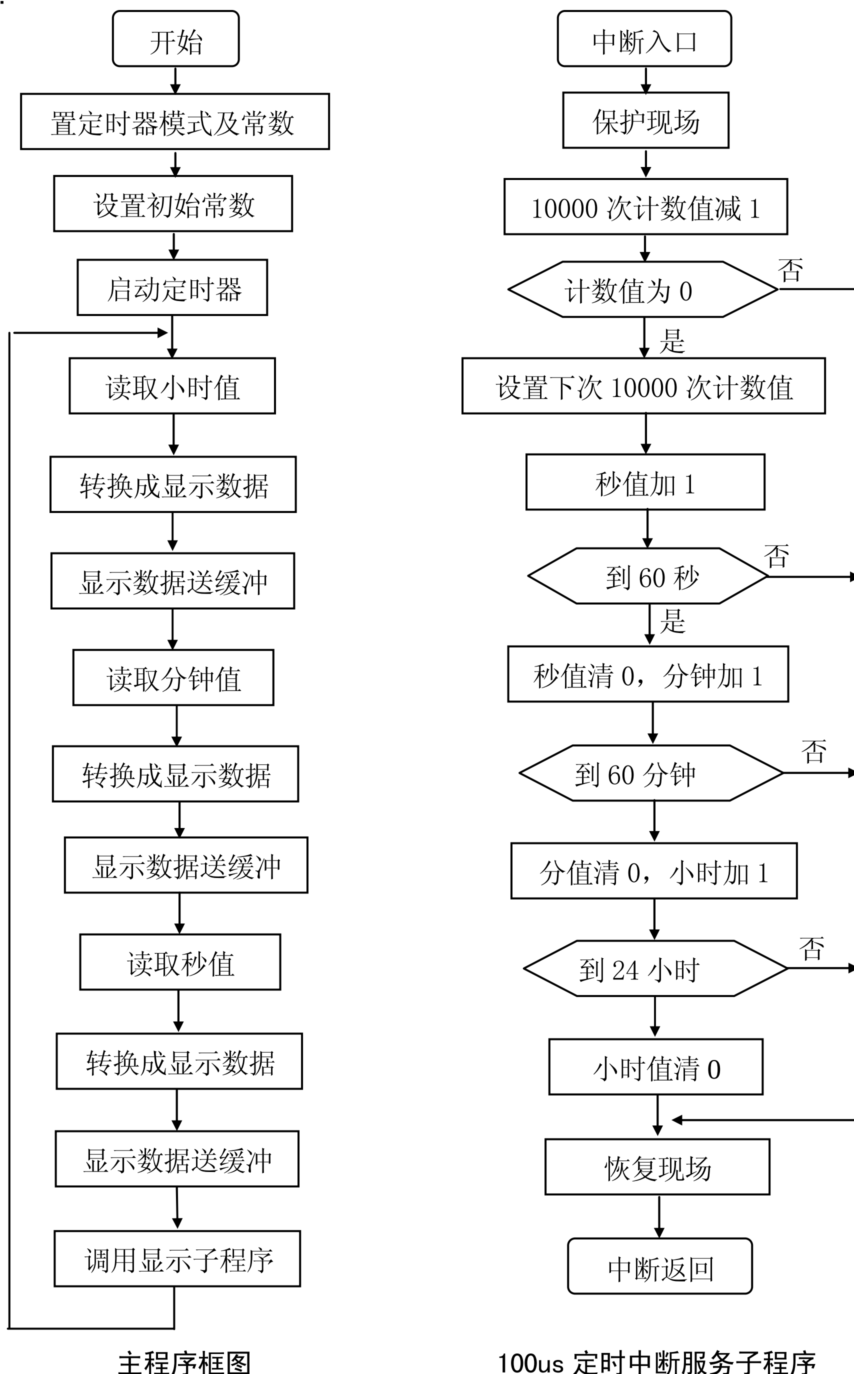
把P3.4 孔分别与“脉冲源”的 2MHZ、1MHZ、0.5MHZ孔相连时,显示值反而比连 0.25MHZ孔更慢,为什么?当f_{osc}=6MHZ时,能够计数的脉冲信号最高频率为多少?

实验十四　电子时钟（定时/计数器定时实验）

1. 实验目的:

进一步掌握定时器、中断处理程序的编程方法；进一步掌握数码显示电路的驱动方法。

2. 实验框图:



主程序框图　　　　100us 定时中断服务子程序

3. 实验内容:

利用 CPU 的定时器和实验仪上提供的数码显示电路，设计一个电子时钟。

4. 实验说明:

MCS-51 单片机有两个 16 位的定时计数器 T0 和 T1，当工作在定时方式时，计数脉冲来自单片机的内部，振荡器信号 12 分频后作计数脉冲，每个机器周期的时间使计数器加 1，由于计数脉冲的频率是固定的（即每个脉冲为 1 个机器周期的时间），故可通过设定计数值来实现定时功能。如要实现定时 100μS，当晶振为 6M 时，每个机器周期为 2μS，让 T0 计 50 个数后溢出，产生中断告诉 CPU 定时，100μS 时间到。本实验定时器每 100us 中断一次，在中断服务程序中,对中断次数进行计数，100us 计数 10000 次就是 1 秒。然后再对秒计数得到分和小时值，并送入显示缓冲区。

5. 实验器材：

（1）G2010+实验平台 1 台；　（2）G6W 仿真器 1 台；　（3）计算机 1 台

6. 实验电路:

见第一章的 1.1.15 “8155 键显模块”

7. 调试方法：

本实验为实时 I/O 实验，因此软件调试一般用全速断点方式运行。将断点分别设在定时器中断入口地址上、中断处理程序结束处、主程序中。 分别以全速断点运行方式开始运行，若都能碰到断点，则说明程序的总体结构是正确的。

8. 程序清单:(MCS51\A14.ASM)

实验十五　INT0 中断实验

1.实验目的：

掌握 MCS-51 单片机中断原理以及编程使用方法；理解下降沿中断和低电平中断的区别。

2.实验内容:

编写主程序，读取 K0 状态，当其与地端闭合时（P1.3 为低电平）初始化为下降沿中断，反之，初始化为低电平中断，且发光二极管灭；编写中断服务程序，使图中的发光管闪烁 5 次，间隔 250ms，即中断服务程序的执行时间为 2.5 秒，退出中断程序时，使发光管灭。

3.实验器材：

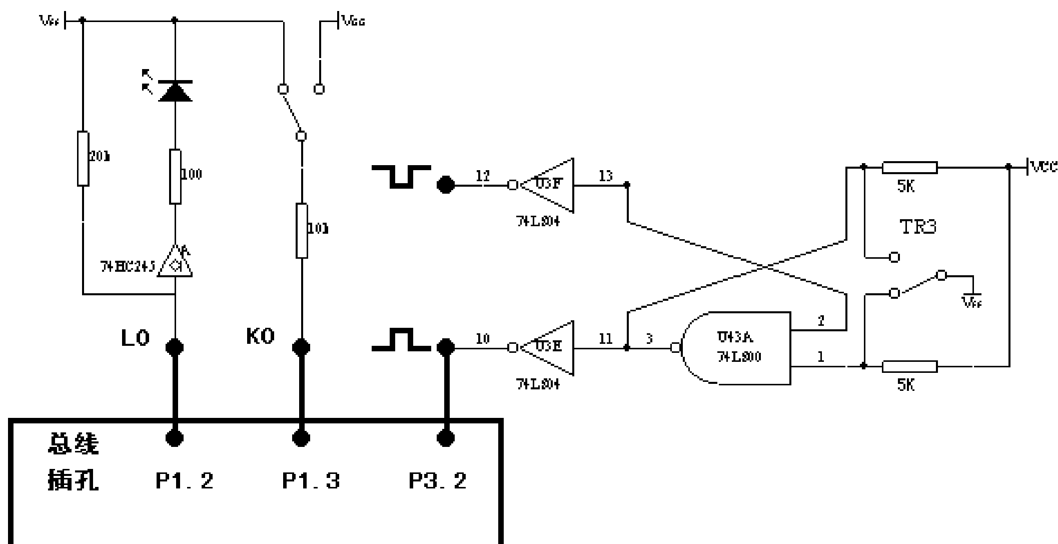
（1）G2010+实验平台 1 台 （2）G6W 仿真器 1 台 （3）连线 若干 根（4）计算机 1 台

4.实验原理：

MCS-51 单片机有 5 个中断源，其中两个是由/INT0、/INT1 引脚输入的外部中断源；另外三个是内部中断源即由 T0、T1 的溢出引起中断和串行口发送完一个字节或接收到一个字节数据引起中断。触发外部中断有两种方式，即下降沿引起中断或低电平引起中断。当编程 TCON 中的 ITi 为 1 时，则引起触发的方式为边沿触发方式，反之为低电平触发方式。每个中断源的中断请求能否得到响应要受两级“开关”的控制，即一个总“开关”

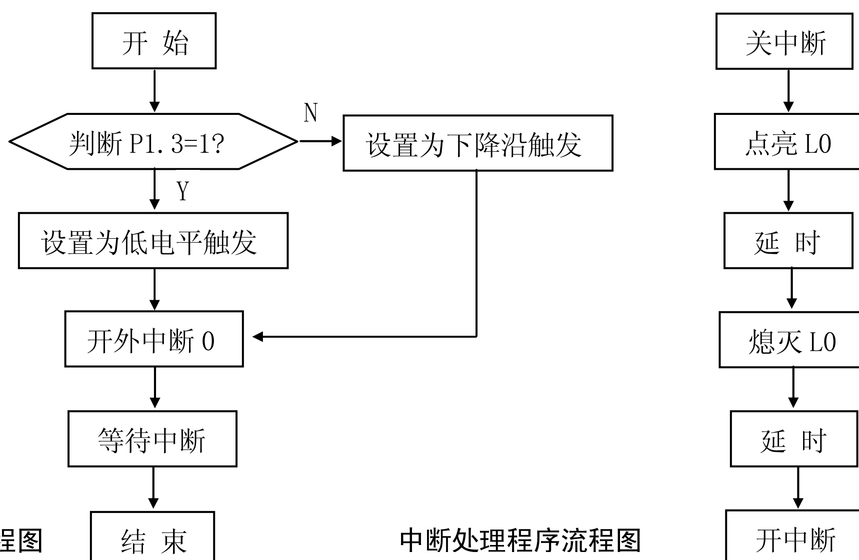
EA 和五个分“开关”ES、ET1、EX1、ET0 和 EX0 的控制，只有当 EA 为 1，且对应的分“开关”也为 1 时，相应中断源的中断请求才能被响应。5 个中断源可编程为两个优先级，同一优级内以/INT0 的优先级最高，然后是 T0 中断、/INT1 中断、T1 中断和串行口中断。5 个中断源对应有个固定的中断服务程序入口地址。CPU 在每个机器周期顺序采样每个中断源，如查询到某个中断源的中断标志为 1，中断系统通过硬件自动将相应的中断矢量装入 PC，进入中断服务程序。图中的开关 TR3 为一带锁按钮（在实验仪“脉冲源”模块中），在按下和松开按钮时，“ ┐┌ ”孔的电平将发生变化，由此产生下降沿。

5.接线方案:



把“总线插孔”框中的 P1.2 孔,P1.3 孔,P3.2 孔分别连“发光二极管组”的 L0 孔，“开关量发生器”的 K0 孔、“脉冲源”的“⊓⊔”孔。

6. 程序框图:



7. 实验步骤:

（1）把仿真器和 G2010 实验平台连接好，编写程序，并运行。

（2）首先将 K0 与地端相连，按按钮 TR3，使图中⊓⊔点变成高电平,等 2 秒后再按按钮 TR3，使图中⊓⊔点为低电平,重复以上操作,观察发光管的闪烁情况。

(3) 然后将 K0 与 VCC 端相连，重新运行程序，重复上述的动作，观察发光管闪烁情况。两次的观察结果是否一样，为什么？

（4）用万用表测量图中⊓⊔点的电平，按下 TR3，然后松开，观察该点电平的变化。

8. 程序清单：（MCS51\A15.ASM）

实验十六　　A/D 转换实验

1.实验目的:

掌握 A/D 转换与单片机接口的方法；了解 A/D 芯片 0809 转换性能及编程方法；

2.实验内容:

利用实验平台上的 0809 做 A/D 转换器，实验平台上的电位器提供模拟量输入，编制程序，将模拟量转换成数字量，通过 8155 键显区数码管显示出来。

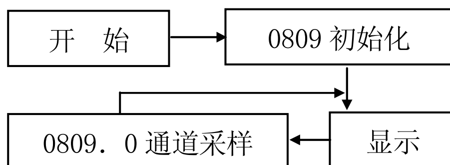
3.实验器材:

（1）G2010+实验平台 1 台（2）G6W 仿真器 1 台 （3）连线 若干 根（4）计算机 1 台

4.实验原理:

A/D 转换器的功能主要是将输入的模拟信号转换成数字信号，如电压、电流、温度测量等都属于这种转换。本实验中采用的转换器为 ADC0809，它是一个 8 位逐次逼近型 A/D 转换器，可以对 8 个模拟量进行转换，转换时间为 100μS。其工作过程如下：首先由地址锁存信号 ALE 的上升沿将引脚 ADDA、ADDB 和 ADDC 上的信号锁存到地址寄存器内,用以选择模拟量输入通道；START 信号的下降沿启动 A/D 转换器开始工作；当转换结束时，AD0809 使 EOC 引脚由低电平变成高电平，程序可以通过查询的方式读取转换结果，也可以通过中断方式读取结果。CLOCK 为转换时钟输入端，频率为 100KHz-1.2MHz，推荐值为 640KHz。

5.程序框图：



6.实验步骤:

(1) 设定工作模式为模式 2，即程序空间在仿真器上，数据空间在用户板上。把 AD0809 的零通道 IN0 孔用连线接至模拟信号发生器的 Vin0 孔，AD0809 的片选信号 CS4 孔接“译码器”YC2（0A000-0AFFFH）孔,“脉冲源”中的 0.5MHZ 孔连 AD0809 的 CLOCK 孔。

（2）硬件诊断：调整实验平台上模拟信号发生器的电位器，使输入到此 AD0809 的 IN0 上电压为一定值。

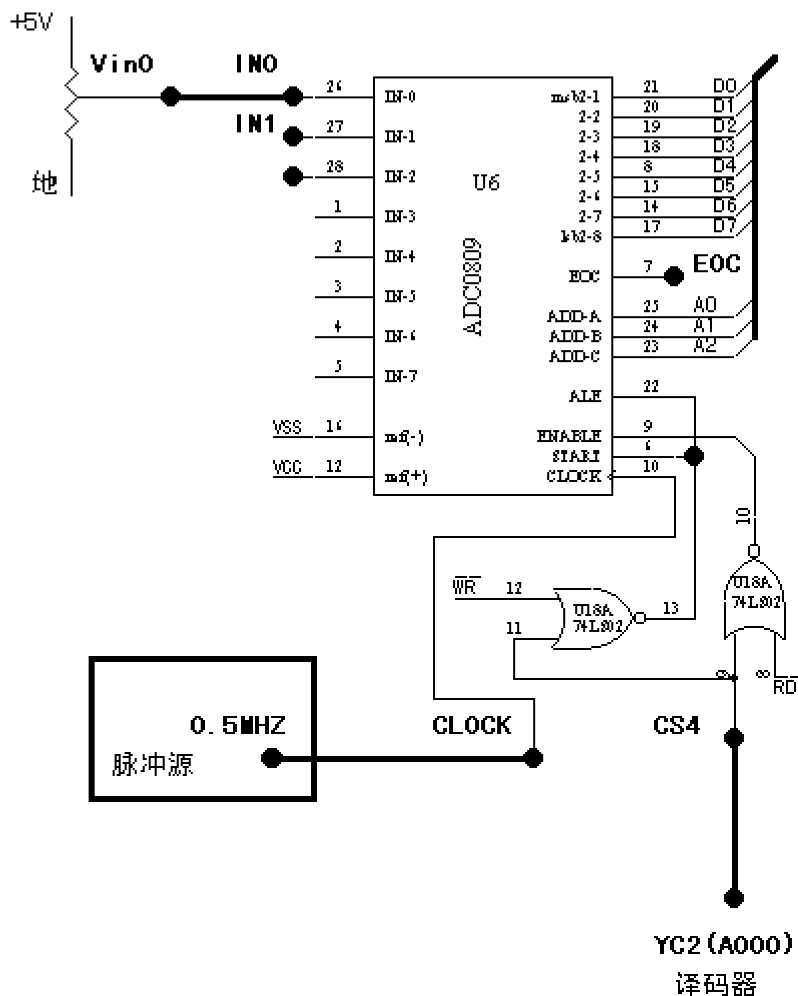
（2．0） G2010+G6W 连 PC 机，在 WINDOWS 调试环境下点击外部数据窗口图标（ ），在打开的 XDATA 窗口中的 0A000H 地址上写入 00H（即选择 AD0809 的 0 通道)，以启动 AD0809 对 IN0 上输入电压进行 A/D 转换，然后点击刷新图标（ ），读出 AD 转换结果。

（2．1） G2010+G6W 连 PC 机，在 DOS 调试环境下，把光标移入 XDATA 窗中，按 Alt+G，在打开的“Goto　Address”窗口中输入“0A000H”，再按“回车”键，在 0A000H 地址上打入“00H”以启动 AD0809 对 IN0 上输入电压进行 A/D 转换，然后屏幕上显示的值即是读出的 AD 转换结果。

（3）编写程序，并编译通过 。本程序使用查询的方式读取转换结果。在读取转换结果的指令后设置断点，运行程序，在断点处检查并读出 A/D 转换结果，数据是否与 Vin0

相对应。修改程序中错误，使显示值随 Vin0 变化而变化。

7. 接线图案：



8.思考问题：

(1)试编写循环采集 8 路模拟量输入 AD 转换程序；(2)以十进制方式显示。

9. 软件清单：（MCS51\A16.ASM）

实验十七　　D/A0832 转换实验

1.实验目的:

了解 D/A 转换与单片机的接口方法；了解 D/A 转换芯片 DA0832 的性能及编程方法。

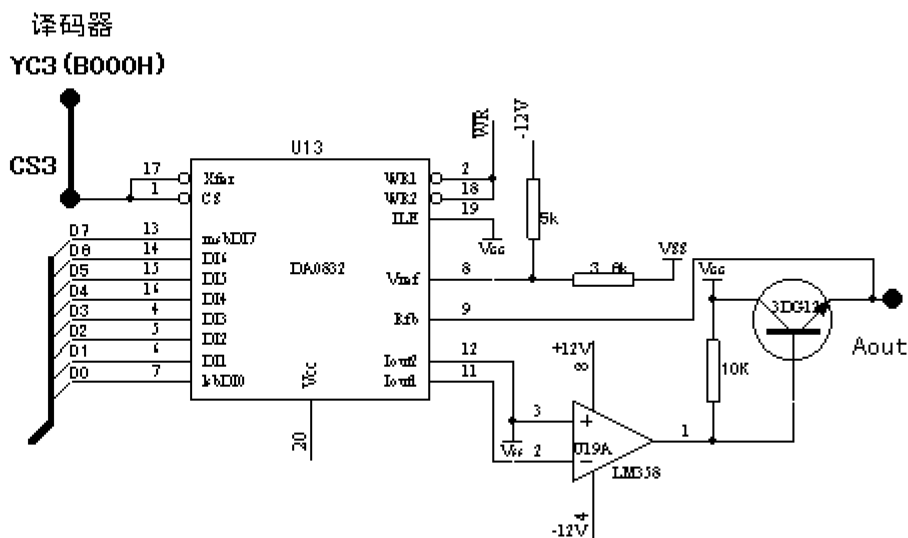
2.实验内容:

利用 0832 输出一个从 0V 开始逐渐升至 5V 再降至 0V 的三角波电压,数码管显示数字量值。

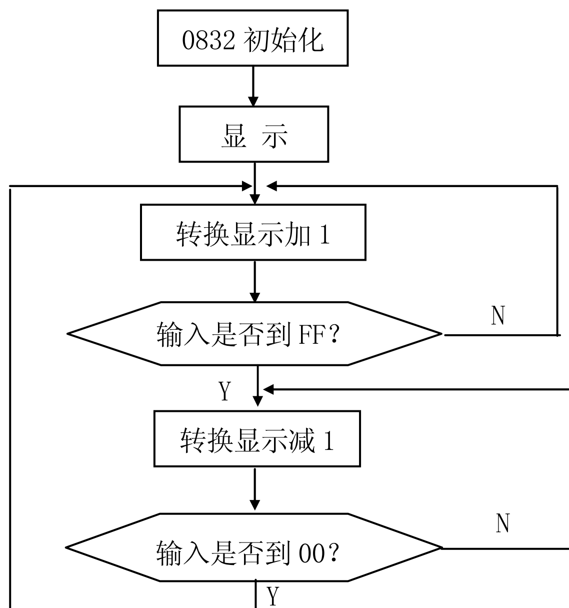
3.实验器材:

（1）G2010+实验平台 1 台（2）G6W 仿真器　1 台　（3）连线　若干 根（4）计算机 1 台

4.接线图案：



5.程序框图：



6.实验原理：

D/A 转换器的功能主要是将输入的数字量转换成模拟量输出，在语音合成等方面得到了广泛的应用。本实验中采用的转换器为 DAC0832，该芯片为电流输出型 8 位 D/A 转换器，输入设有两级缓冲锁存器，因此可同时输出多路模拟量。本实验中采用单级缓冲连接方式，用 0832 来产生三角波，具体线路如上图所示。Vref 引脚的电压极性和大小决定了输出电压的极性与幅度，G2010+实验平台上的 DA0832 的第 8 引脚（Vref）的电压已接为-5V，所以输出电压值的幅度为 0-5V。

7. 实验步骤：

（1）设定工作模式为模式 2，即程序空间在仿真器上，数据空间在用户板上。把 DA0832 的片选 CS3 孔接至 0B000H-0BFFFH 孔。

（2）硬件诊断：

（2．0） G2010+G6W 连 PC 机，在 WINDOWS 调试环境下点击外部数据窗口图标（ ），在打开的 XDATA 窗口中的 0B000H 地址（即 DA0832 的片选空间）上写入 FFH，则 Aout 孔输出应为 5V，将 00H 写入，则 Aout 孔输出应为 0V。

（2．1） G2010+G6W 连 PC 机，在 DOS 调试环境下，把光标移入 XDATA 窗中，按 Alt+G，在打开的“Goto Address”窗口中输入“0B000H”，再按“回车”键，在 0B000H 地址上打入“0FFH” 则 Aout 孔输出应为 5V，将 00H 写入，则 Aout 孔输出应为 0V。

（2．2）如选购了 G2K 仿真板，则还可直接在 G2010+实验平台的键盘上进行硬件诊断：设定工作模式为模式 2，即数码管显示为“PI EE”，然后按“MON”键使数码管显示为“ ’ ”，输入“B000”，按“+”，再输入“FF”，则 Aout 孔输出应为 5V，将 00H 写入，则 Aout 孔输出应为 0V。

（3）编写程序、编译程序，用单步、断点、连续方式调试程序，排除软件错误。运行程序，8155 键显区数码管上显示不断加大或减小的数字量，用万用表测量 D/A 输出孔 AOUT，应能测出不断加大或减少的电压值。

8. 思考问题：

修改程序，使能产生锯齿波。

9. 软件清单： （ MCS51\A17.ASM)

实验十八　　电子琴

1.实验目的：

了解发出不同音调声音的编程方法。

2.实验内容：

利用实验仪上提供的键盘，使数字键 1、2、3、4、5、6、7 作为电子琴按键，按下即发出相应的音调。用 P3.0 口发出音频脉冲,驱动喇叭.。

3. 实验原理：

我们知道，声音是由振动产生的，每个音符都对应了一个频率如表 1 所示。利用定时/计数器 T0 工作在 16 位定时方式，通过改变 TH0 和 TL0 的值，就可以产生不同频率的脉冲，例如想产生 523Hz（音符 1 的发音）的脉冲，其周期为 1/523=1912μS，因此只要让 T0 定时 956μS 后，使 P3.0 取反，就可以在 P3.0 引脚上输出一个频率为 523Hz 的脉冲。若晶振的频率为 6MHz，则计数值为 956/2=478，而计数器的初值为 65536-478=65058=0FF22H，即 TH0=0FFH，TL0=22H。这样每个音符都对应了一个 T 值，6M 晶振时各音符的 T 值如下表：

音符频率以及 6M 晶体时对应的 T 值表

音符	频率	T 值	音符	频率	T 值
$\underset{\cdot}{1}$	262	64582	1	523	65058
$\underset{\cdot}{2}$	294	64685	2	578	65110
$\underset{\cdot}{3}$	330	64778	3	659	65156
$\underset{\cdot}{4}$	349	64819	4	698	65178
$\underset{\cdot}{5}$	392	64898	5	784	65217
$\underset{\cdot}{6}$	440	64968	6	880	65252
$\underset{\cdot}{7}$	494	65030	7	988	65283

另一方面是每个音符的发音长度，各调节拍与时间的设定如下表所示：

调值与节拍延时时间关系表

曲调值	1/4 拍时间	1/8 拍时间
调 4/4	125ms	62ms
调 3/4	187ms	94ms
调 2/4	250ms	125ms

4.实验器材:

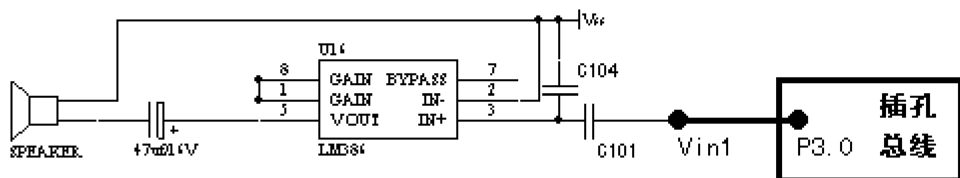
（1）G2010+实验平台　　1　　台

（2）G6W 仿真器　　1　　台

（3）连线　　若干　根

（4）计算机　　1　　台

5.接线图案：



6.实验步骤:

把 P3.0 用连线连至“音响与合成”框 LM386 的 Vin1 插孔上。

7.程序清单： （MCS51\A18.ASM）

实验十九　步进电机控制

1.实验目的：

了解步进电机工作原理，掌握用单片机的步进电机控制系统的硬件设计方法，熟悉步进电机驱动程序的设计与调试，提高单片机应用系统设计和调试水平。

2.实验内容：

编写并调试出一个实验程序按下图所示控制步进电机旋转：

3. 工作原理：

步进电机是工业过程控制及仪表中常用的控制元件之一，例如在机械装置中可以用丝杠把角度变为直线位移，也可以用步进电机带螺旋电位器，调节电压或电流，从而实现对执行机构的控制。步进电机可以直接接收数字信号，不必进行数模转换，用起来非常方便。步进电机还具有快速启停、精确步进和定位等特点，因而在数控机床、绘图仪、打印机以及光学仪器中得到广泛的应用。

步进电机实际上是一个数字/角度转换器，三相步进电机的结构原理如图所示。从图中可以看出，电机的定子上有六个等分磁极，A、A′、B、B′、C、C ′，相邻的两个磁极之间夹角为60°，相对的两个磁极组成一相（A-A′，B-B′，C-C′），当某一绕组有电流通过时，该绕组相应的两个磁极形成N极和S极，每个磁极上各有五个均匀分布矩形小齿，电机的转子上有40个矩形小齿均匀地分布的圆周上，相邻两个齿之间夹角为9°。

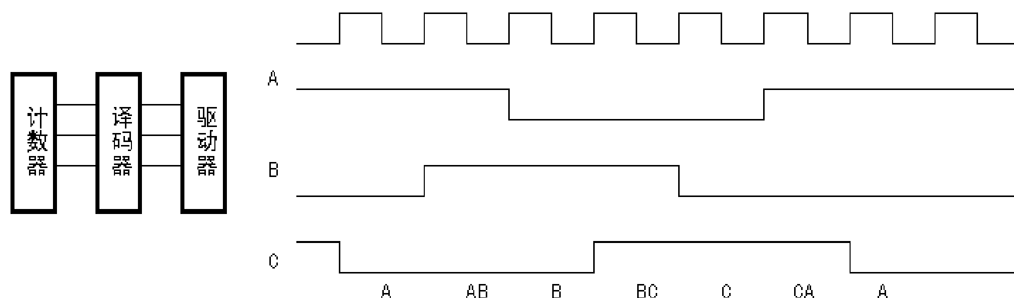
当某一相绕组通电时，对应的磁极就产生磁场，并与转子形成磁路，如果这时定子的小齿和转子的小齿没有对齐，则在磁场的作用下，转子将转动一定的角度，使转子和定子的齿相互对齐。由此可见，错齿是促使步进电机旋转的原因。

三相步进电机结构示意图

例如在三相三拍控制方式中，若A相通电，B、C相都不通电，在磁场作用下使转子齿

和 A 相的定子齿对齐，我们以此作为初始状态。设与 A 相磁极中心线对齐的转子的齿为 0 号齿，由于 B 相磁极与 A 相磁极相差 120°，不是 9°的整数倍（120÷9=40/3），所以此时转子齿没有与 B 相定子的齿对应，只是第 13 号小齿靠近 B 相磁极的中心线，与中心线相差 3°，如果此时突然变为 B 相通电，A、C 相不通电，则 B 相磁极迫使 13 号转子齿与之对齐，转子就转动 3°，这样使电机转了一步。如果按照 A→B→C 的顺序轮流通电一周，则转子将动 9°。

步进电机的运转是由脉冲信号控制的，传统方法是采用数字逻辑电路——环形脉冲分配器控制步进电机的步进。 下图为环形脉搏冲分配器的简化框图。



三相六拍环形脉搏冲分配器

①运转方向控制。如图所示，步进电机以三相六拍方式工作，若按 A→AB→B→BC→C→CA→A 次序通电为正转，则当按 A→AC→C→CB→B→BA→A 次序通电为反转。

②运转速度的控制。图中可以看出，当改变 CP 脉冲的周期时，ABC 三相绕组高低电平的宽度将发生变化，这就导至通电和断电的变化的速率发生了变化，使电机转速改变，所以调节 CP 脉冲的周期就可以控制步进电机的运转速度。

③旋转的角度控制。因为每输入一个 CP 脉冲使步进电机三相绕组状态变化一次，并相应地旋转一个角度，所以步进电机旋转的角度由输入的 CP 脉冲数确定。

G2010 实验仪选用的是 20BY-0 型 4 相步进电机，其工作电压为 4.5V,在双四拍运行方式时,其步距角为 18^0,相直流电阻为 55Ω,最大静电流为 80Ma。采用 8031 单片机控制步进电机的运转，按四相四拍方式在P1 口输出控制代码，令其正转或反转。因此P1 口输出代码的变化周期T控制了电机的运转速度：

$$n=60/T.N$$

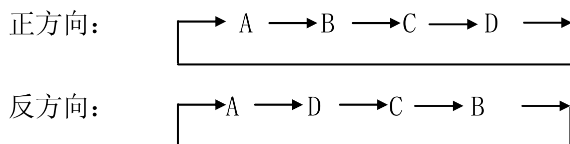
式中：n —— 步进电机的转速（转/分）;

N —— 步进电机旋转一周需输出的字节数;

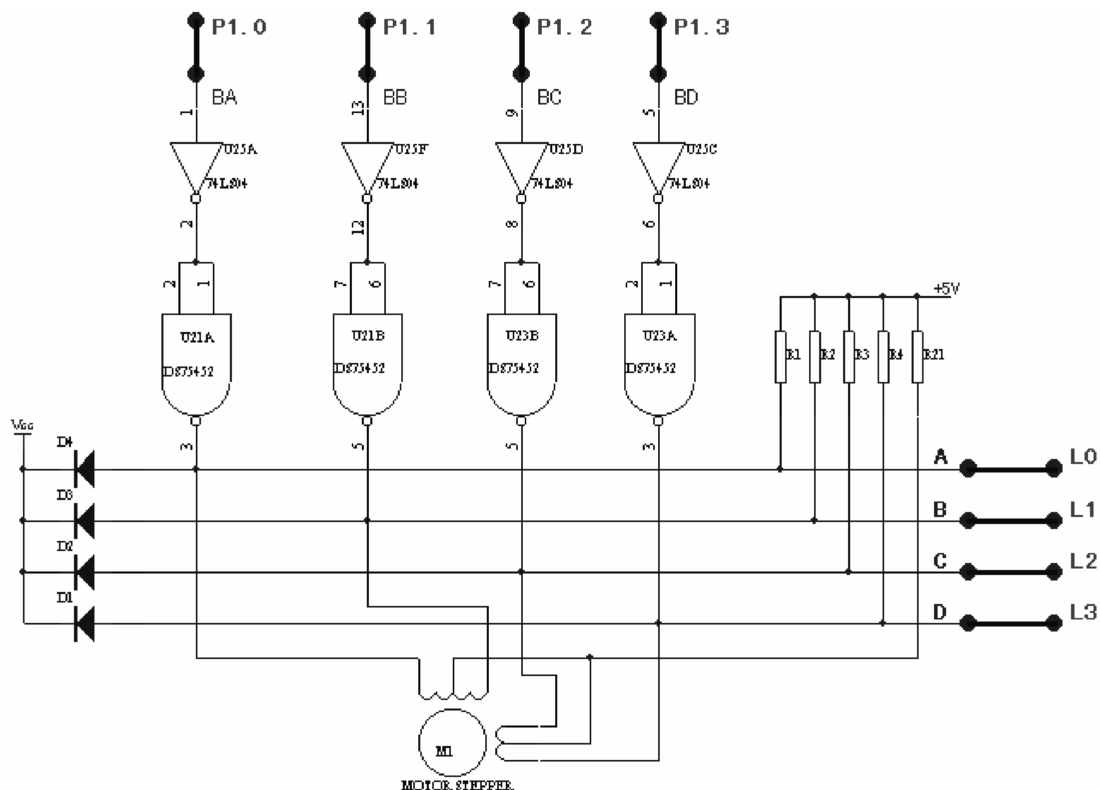
T —— 代码字节的输出变化周期。

设 N=360°/ 18°=20，T=1.43ms,则步进电机的转速为 2100 转/分。

控制 P1 口输出的代码字节个数即控制了步进电机的旋转角度。



4. 接线图案：



根据步进电机工作原理，使用 8031 的 P1.0-P1.3 分别驱动步进电机 A、B、C、D 相，用软件控制 P1 口输出一脉冲序列，控制步进电机转速、方向、步距。同时为能观察步进电机旋转状态，在 A、B、C、D 相输出到状态指示灯。

5. 实验步骤：

（1）**“总线插孔”区的** P1.0-P1.3 孔接步进电机的 BA-BD 孔，“发光二极管组”的 L0-L3 孔接步进电机 A、B、C、D 孔 。P1.7 孔连 L7。

（2）硬件诊断：

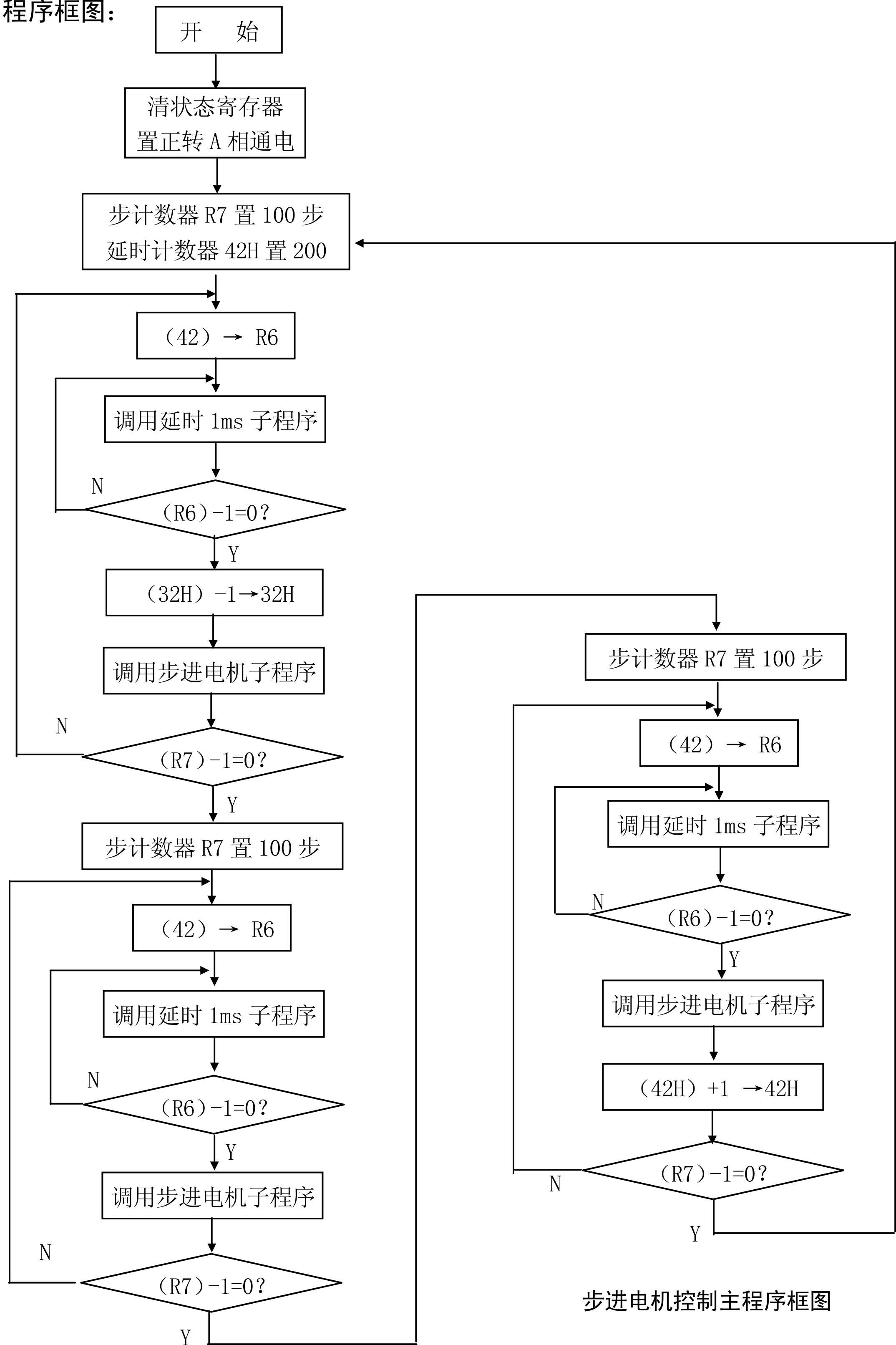
（2．0） G2010+G6W 连 PC 机，在 WINDOWS 调试环境下点击 CPU 窗口图标（），打开 CPU 窗口，对 P1 口写入 55H 或 AAH，则 A 相、B 相、C 相、D 相四只状态指示灯分别正确对应写入的数据。对 P1 口依次写入 0F3H、0F6H、0FCH、0F9H，则应使步进电机正转四步。对 P1 口依次写入 79H、7CH、76H、73H，则应使步进电机反转四步。

（2．1） G2010+G6W 连 PC 机，在 DOS 环境下，在 CPU 窗口中重复以上对 P1 口操作。

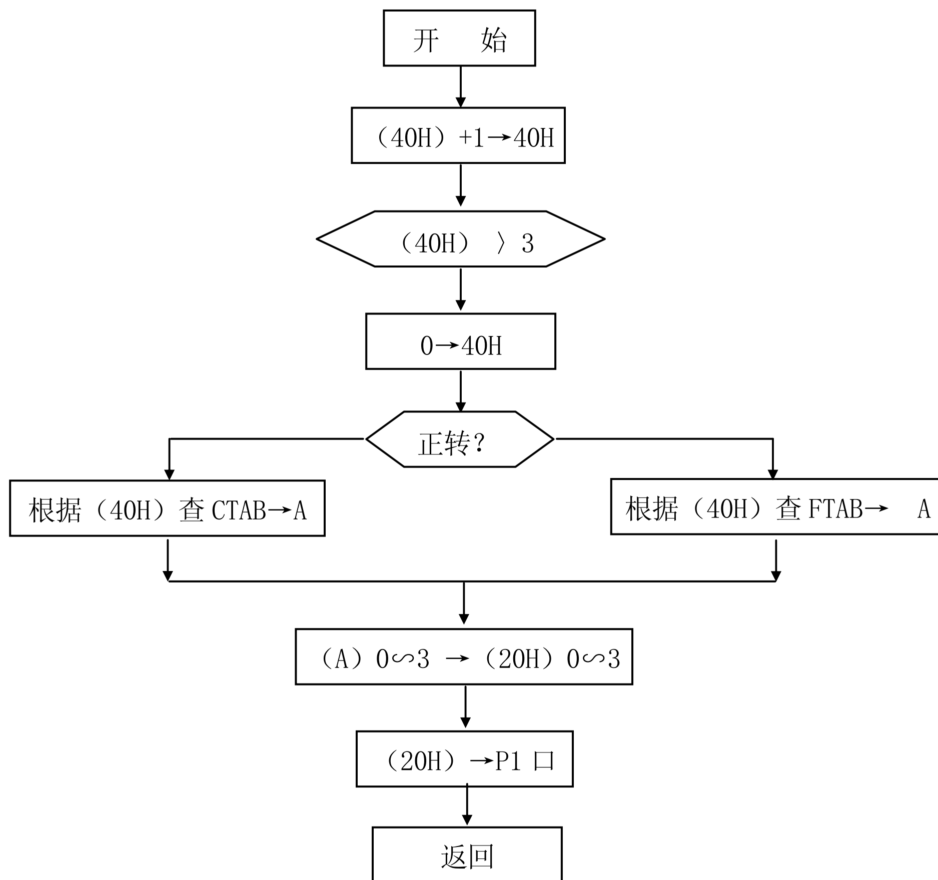
（2．2） 如选购了 G2K 仿真板，则还可直接在 G2010+实验平台的键盘上进行硬件诊断：在键盘上输入“90”（即 P1 口地址），按“+”键后输入“55”、“AA”等数据，看 LED 及步进电机状态变化。

（3）编写程序、编译程序。用单步、全速断点、连续方式调试程序，观察数码管上数字变化，检查程序运行结果，观察步进电机的转动状态，连续运行时用示波器测试 P1 口的输出波形，排除软件错误，直至达到本实验的设计要求。

6. 程序框图：



步进电机控制主程序框图



正反转步进子程序框图

7. 思考问题：

（1）若将步进电机 A、B、C、D 相分别接到 P1.4 -- P1.7，软件功能与本实验要求一致，需要修改那几处程序？

8. 软件清单：(MCS51\A20.ASM）

实验二十　数据存储器 6264RAM 实验

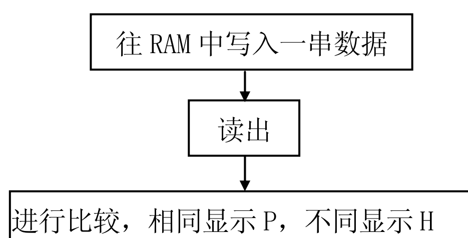
1.实验目的：

学习 RAM6264 的扩展

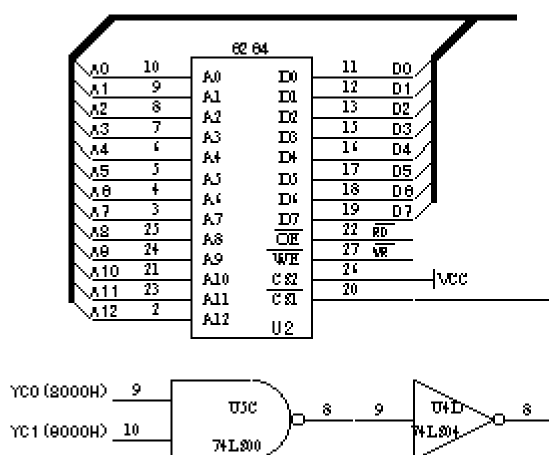
2.实验内容：

往 RAM 中写入一串数据，然后读出，进行比较。

3.程序框图：



4.接线图案：



5.实验步骤：

设计程序，运行并观察结果。

6.软件清单 (MCS51\A20.ASM)

实验二十一　EPROM 固化及脱机运行

1.实验目的：

掌握调试步骤的最后一环：EPROM 固化程序的方法，以及进行脱机运行实验。

2.实验内容：

把先前做的“电子时钟”实验的程序固化到 EPROM 27C256 之中，观察脱机运行情况。

3.实验器材：

（1）　通用编程器　　1　　台

（2）　27C256 芯片　　1　　片

（3）　G2010+实验平台　1　　台

4.实验步骤：

（1）　请另备 27C256 芯片。把 U14 上原有的芯片拔下。

（2）　编程器的使用方法，请参考有关编程器使用手册。

（3）　编程、编译程序，把编译通过的程序的目标码存盘。

（4）　使用编程器把目标码固化在 27C256 之中。

（5）　把 27C256 插入 U14 插座上，把把仿真器与 Lab51 板分离，然后上电。

（6）　观察脱机运行情况。

5.程序清单：（MCS51\A21.ASM）

实验二十二　逻辑分析仪在教学/实验中的利用

1.实验目的:

在教学实践中，时序关系的讲解是一个难于讲透的问题，原因在于学生无法感受到时序关系与指令系统之间的因果关系。而如果使用逻辑分析仪与仿真器的交互调试功能，通过计算机投影机，则讲解起来轻松的多。

2.实验内容:

通过下面的程序，我们用逻辑分析仪观察一个写周期及一个串行口数据的发送. 为了叙述方便,我们在前几条语句后加上了该条指令的地址及机器码。

```
                ORG      0000H
        MOV     DPTR, #1234h    ; 0000  90 12 34
        MOV     A, #56h         ; 0003  74 56
        MOVX    @DPTR, A        ; 0005  F0
        NOP                     ; 在这行设第一个断点
        MOV     TMOD,#020H      ; TIMER1: 方式 2 (8 位重载)
        MOV     TH1,#0F3H       ; 2400BPS @ 12MHz
        MOV     TL1,#0F3H
        ANL     PCON,#07FH      ; 清 SMOD 位
        MOV     SCON,#50H
        SETB    TR1
        MOV     A, #055H
        MOV     SBUF, A         ; 发送 55H
Wait:
        JNB     TI, Wait
        SJMP    $               ; 在这行设第二个断点
        END
```

3.实验器材:

（1）E2000/L 仿真器　　1　　台　　（2）逻辑勾　　1　　台

4.实验步骤:

（1）在第五行及第十七行设置两个断点

（2）首先设置采样频率 20M, 全速执行到第一个断点, 这时逻辑分析仪采样到为前三条指令的执行波形, 下页我们会结合波形详细说明.接下来我们来采样串行口发送出的数据波形, 我们在程序中设置的波特率为 2400, 每一个数据位的长度应为:

1000000us/2400 = 416.667us

串行口的发送数据的格式为一个起始位 0, 8 个数据位(低位在前, 高位在后),一个停止位 1,共十位. 我们在程序中发送了 55H, 因此串行口数据为:

0,　1,0,1,0,1,0,1,0,　1

我们设置采样频率为1M, 选择逻辑分析仪A组由外部J3引入, 将J3第0位测试钩(红色)钩在仿真头第11脚(TXD引脚)上, 全速执行到第二个断点. 看下页波形。

下图显示的是 MOVX @DPTR, A 的执行波形, 这条指令的地址为 0005, 机器码为F0.图中AH, AL, P0分别为高位地址, 低位地址, P0口.

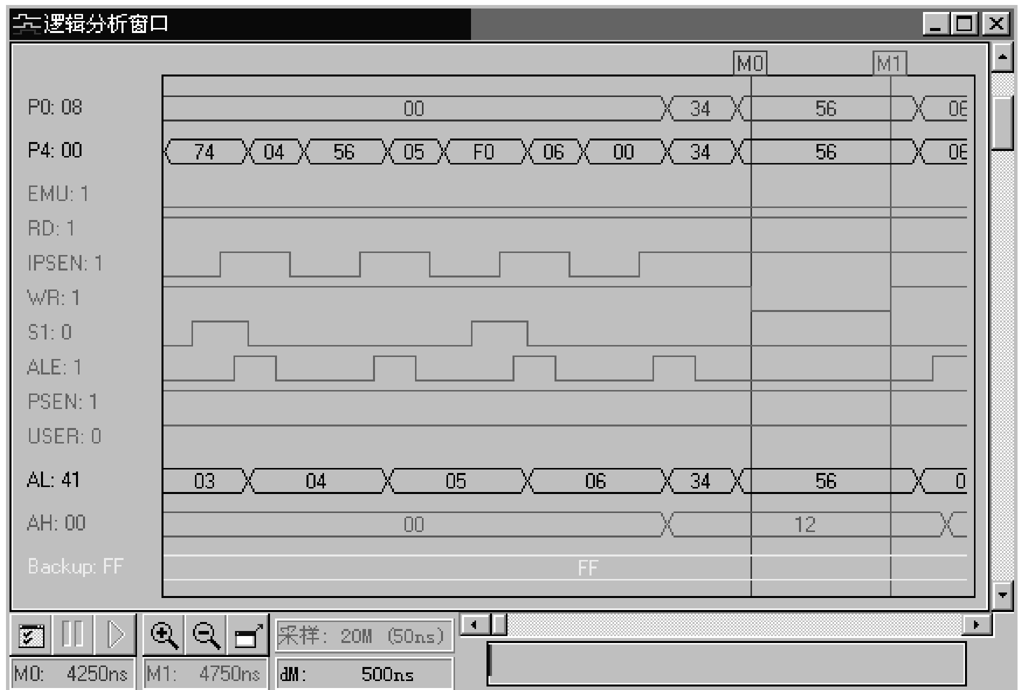
A. ALE 的上升沿后P0口输出地址05, ALE下降沿, 地址被锁存.

B. 然后, PSEN变低, 开始读指令, 稍后机器码F0被读出.

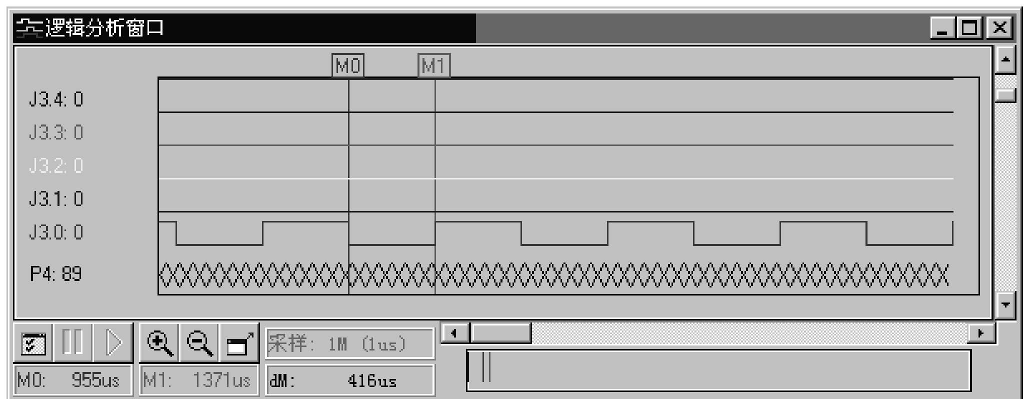
C. 接下来读入的0006单元00 为MCS51 CPU 取的下一条指令, 被丢弃

D. CPU 送出要访问单元地址 1234H, 接着送出A的内容56H, 并输出写信号

E. 将M0, M1分别移至WR信号上升/下降沿. 此时dM: 显示的为WR的宽度为500ns



(3) 观察结果:



上图显示的是串行输出脚上的输出波形. 可以看出输出的串口数据为: 0,1,0,1,0,1,0,1,0,1 最后一个停止位 1 未输出完是因为 MCS51 CPU 在开始发送停止位后就置位 TI 的原故.移动 M0, M1 可测出一个数据位的宽度为 dM: 416us, 并与理论数据 416.667us 相吻合.

5. 程序清单:（MCS51\A22.ASM）

第❷节　MCS51“模仿式”实验例程

实验一　工业顺序控制（INT0、INT1）综合实验

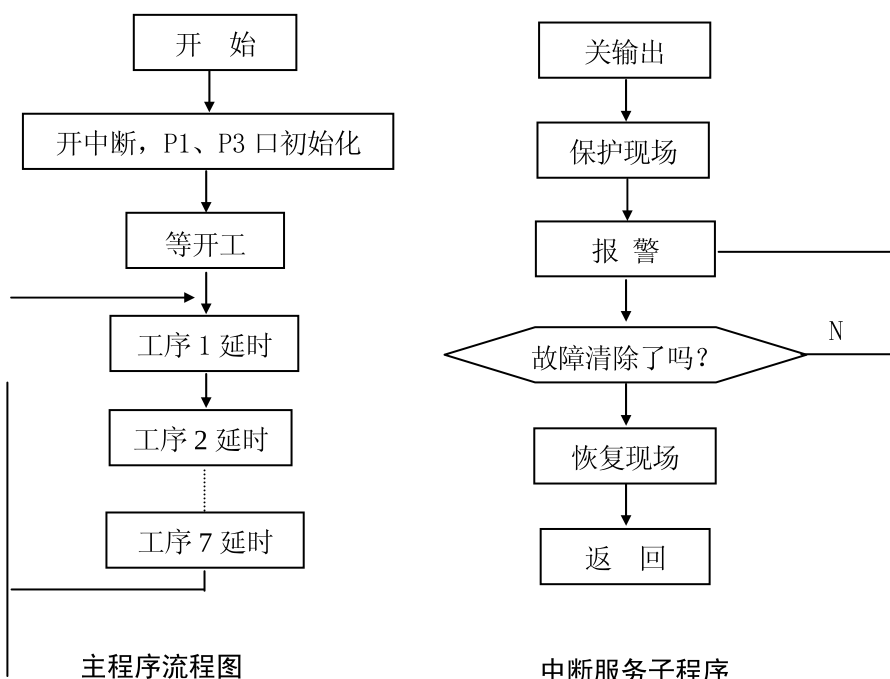
1.实验目的：

掌握工业顺序控制程序的简单编程，中断的使用。

2.实验内容：

8031 P1.0-P1.6 控制注塑机七道工序，现模拟控制七只发光二极管的点亮，高电平点亮。设定每道工序时间转换为延时，P3.4 为开工启动开关，高电平启动。P3.3 为外部故障输入模拟开关，低电平报警，P1.7 为报警声音输出。设定 7 道工序只有一位输出。

3.程序框图：



主程序流程图　　　　中断服务子程序

4.接线图案：

5. 实验步骤：

按图接好连线。执行程序，把 K0 接到高电平，观察发光二极管点亮情况，确定工序执行是否正常，然后把 K1 置为低电平，看是否有声音报警。恢复中断 1，报警停，又从刚才报警时一道程序执行下去。

6. 思考问题：

修改程序，使每道工序中有多位输出。

7. 程序清单：（MCS51\B01.ASM）

实验二　扩展时钟系统实验（DS12887）

1．实验目的：

掌握 MSC-51 单片机扩展时钟电路的设计方法；了解 DS12887 的工作原理。

2．实验内容：

编程实现下列功能：

（1） 程序第一次运行后，初始化时间显示为 00：00：00，即 6 位数码管显示为 00.00.00。通过键盘[MON]设定小时为 07，通过键盘[LAST]设定分钟为 08，通过键盘[NEXT]设定秒为 09，两分钟后即在 7.10.09 时关掉电源，等待 2 分钟后再打开电源，这时时间应为 7.12.09，即停电后 DS12887 中的时钟不会停止运行。

3．实验器材：

（1）G2010+实验平台 1 台（2）G6W 仿真器 1 台 （3）连线 若干 根 （4）计算机 1 台

4．实验原理：

在很多应用场合要求单片机系统不仅能够准确地采集数据，而且还需要了解产生这些数据的时刻，为单片机系统增加日历时钟是一项非常有用的技术，掌握这项技术便是本实验的目的。实验中使用 DALLAS 公司生产的日历、时钟加 RAM 芯片 DS12887。它具有接口简单，使用方便等特点，曾被用在 586 计算机中。其引脚分布如图所示，内部有 128 字节的非易失 SRAM，具体分配也如图所示。

MOT	VCC
NC	SQW
NC	NC
AD0	NC
AD1	NC
AD2	$\overline{IRQ}$
AD3	$\overline{RST}$
AD4	DS
AD5	NC
AD6	$R/\overline{W}$
AD7	AS
GND	$\overline{CS}$

引脚分布图

14 BYTE	00 … 0D
114 BYTE	… FF

00	SECONDS
01	SECODES ALARM
02	MINUTES
03	MINUTES ALARM
04	HOURS
05	HOURS ALARM
06	DAY OF THE WEEK
07	DAY OF THE MONTH
08	MONTH
09	YEAR
0A	REGISER A
0B	REGISER B
0C	REGISER C
0D	REGISER D

存储器分布图

通过对寄存器 A、B、C、D 的编程可以控制 DS12887 的工作方式。

寄存器 A

D7	D6	D5	D4	D3	D2	D1	D0
UIP	DV2	DV1	DV0	RS3	RS2	RS1	RS0

UIP 位当其为 0 时指示更新在 244μS 内不会发生；DV2DV1DV0 当其为 010 时，打开晶振，并允许时钟开始计时；RS3RS2RS1RS0 用于选择周期中断或输出方波频率，当其分别为 0111、1000、1001、1011、1101、1110、1111 时，对应频率为 512Hz、256Hz、128Hz、64Hz、32Hz、16Hz、8Hz、4Hz、2Hz。

寄存器 B

D7	D6	D5	D4	D3	D2	D1	D0
SET	PIE	AIE	UIE	SQW	DM	12/24	DSE

SET 位为 0 时，每秒计数一次，置 1 后，更新转换被禁止；PIE、AIE、UIE 位当它们为 1 时，分别允许周期中断、报警中断和时钟数据更新结束中断，为 0 时，禁止中断产生；SQWE 位当其为 1 时，按以寄存器 A 中由 RS3RS2RS1RS0 设定的频率从 SQW 引脚输出方波，当其为 0 时，SQW 为低电平；当 DM 为 1 时选用二进制数据格式，反之为 BCD 数据格式；12/24 位为 1 时，指定 24 小时时间格式，否则为 12 小时时间格式；DSE 为 1 时允许夏时制发生。

寄存器 C

D7	D6	D5	D4	D3	D2	D1	D0
IRQF	PF	AF	UF	0	0	0	0

寄存器 C 的内容是周期中断标志位 PF、报警中断标志位 AF、更新结束中断标志位 UF 和中断请求标志位 IRQF，它们之间的关系为 IRQF=PF*PIE+AF*AIE+UF*UIE，只要 IRQF 为 1，/IRQ 引脚输出就保持低电平，读寄存器 C 将清除所有标志。

寄存器 D

D7	D6	D5	D4	D3	D2	D1	D0
VRT	0	0	0	0	0	0	0

寄存器 D 中仅 D7 有定义，读时应总为 1，若为 0 则说明内部锂电池已耗尽。

为防止锂电池在芯片装入系统前被耗尽，DS12887 在出厂时先关掉了其内部的晶振，编程时必须首先给寄存器 A 的 DV2DV1DV0 位写入 010 以打开晶振，然后读寄存器 D 以检查内部锂电池是否有效；接着根据需要对寄存器 A、B 进行设置。当需要修改日历时钟时，需要先使 SET 位置 1，当需要读日历时钟数据时，必须先查询寄存器 A 中的 UIP 位，只有当其为 0 时，才能进行读取数据。

5、接线图案：

6．程序框图：

开 始

8155 初始化

DS12887 初始化

读秒、分、时

拆字后送显示缓冲器

调用显示子程序

调用键扫子程序

有键按下吗？

是 NEXT 键？

调整秒位

是 LAST 键？

调整分位

是 MON 键？

调整时位

7．实验步骤：

（1）设定工作模式为模式 2，即程序空间在仿真器上，数据空间在用户板上。

（2） 从“接线图案”中看出，DS12887 的/CS 脚已连接 U17 译码器的 YS7 脚，于是可知 DS12887 的地址空间为 0FE00H-0FFFFH。

（3）硬件调试：因为 DS12887 内部有 114 个内部 RAM，在系统中的地址空间为：0FE0EH-0FE7FH，所以，可通过查看这一区域的数据读写来判断硬件故障否。

（3．0） G2010+G6W 连 PC 机，在 WINDOWS 调试环境下点击外部数据窗口图标（），在打开的 XDATA 窗口中的 0FE10H 开始的地址上写入一串 55H，然后点击刷新图标（）， 如写入的一串 55H 未被修改，则硬件无故障。

（3．1） G2010+G6W 连 PC 机，在 DOS 调试环境下，把光标移入 XDATA 窗中，按 Alt+G，在打开的“Goto Address”窗口中输入“0F010H”，再按“回车”键，在 0F010H

地址上打入一串数，看能否修改。若能修改，则表明硬件无故障。

（3．2） G2010+G2K 仿真板，则还可直接在 G2010+实验平台的键盘上进行硬件诊断：设定工作模式为模式 2，即数码管显示为“PI EE”，然后按“MON”键使数码管显示为“ ’ ”，输入“F010”，按“+”，再输入“AA”， 再按“+”，“-”键，此时，数码管显示的“F010 AA”，则表明硬件无故障。

(4)设计程序并进行调试。

8．程序清单：（MCS51\B02.ASM）

实验三　双机通讯实验

1.实验目的：

（1）掌握单片机串行口工作方式的程序设计及简单三线式通讯的方法。

（2）了解实现串行通讯的硬环境、数据格式的协议、数据交换的协议。

（3）学习串行口通讯的中断方式程序的编写方法。

2.实验内容：

利用 8031 单片机串行口，实现两个实验台之间的串行通讯。其中一个实验台作为发送方，另一侧为接收方。发送方读入按键值，并发送给接收方，接收方收到数据后在 LED 上显示。

3.接线方案：

4.实验步骤：

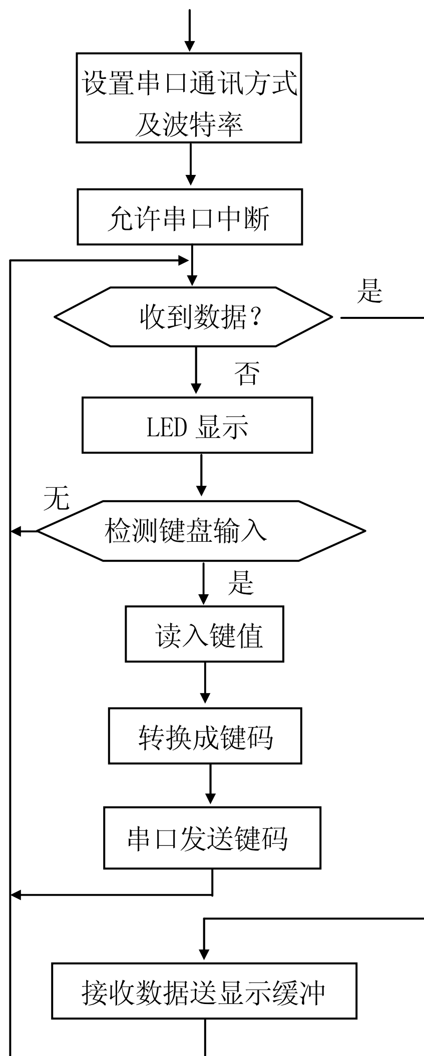
(1)把甲机 RX1、TX1 孔分别连甲机的 P3.0、P3.1 孔，乙机 RX1、TX1 孔也分别连乙机的 P3.0、P3.1 孔。甲机 RS232 插座 3、2 连乙机 RS232 插座 2、3，两机的地线相连。可直接用随机的通信电缆连接。

(2)设计程序，分别在甲机，乙机上运行。然后，在甲机键盘上按下 0-F 键 ,应在乙机上的 8155 键显区数码器上显示相应值。

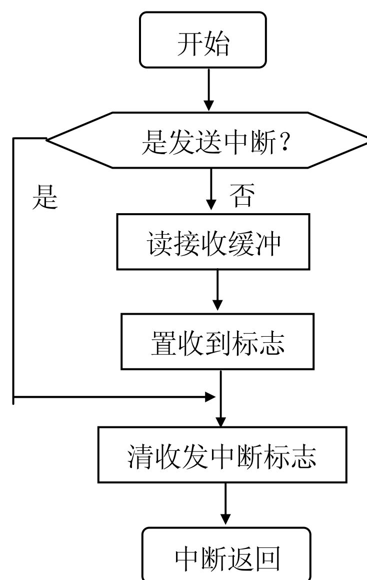
(3)若把 P3.0、P3.1 孔直接相连,则按键值在本机的数码管上显示出来。

5.程序框图：

开始



主程序



串口中断子程序

6. 程序清单：（MCS51\B03.ASM）

实验四　V/F 压频转换实验

1. 实验目的：

了解 LM331 电压转换为频率的基本工作原理，熟悉 8031 内部定时/计数器的使用方法。

2. 实验内容：

把电压转换成脉冲，用计数器进行测频并在 SICElab—G2010+的数码管上显示出来，实现频率计功能。

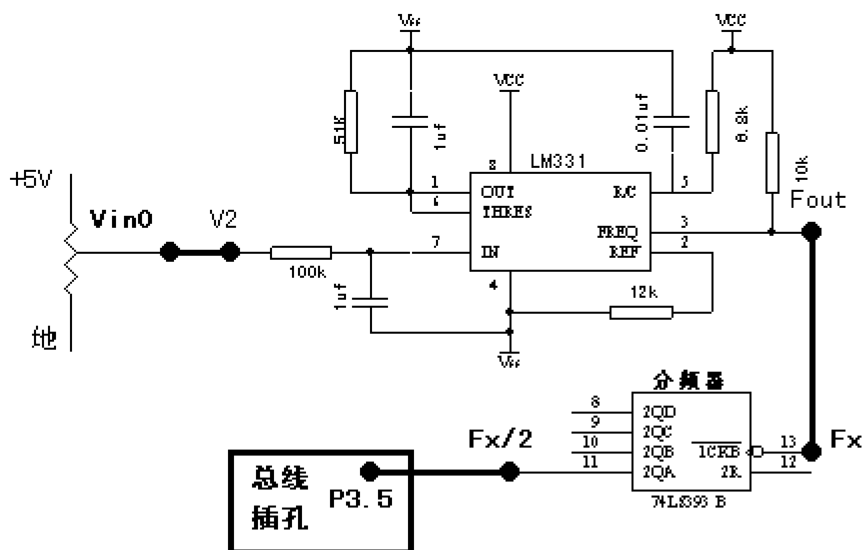
3. 工作原理：

把模拟信号送 LM331 进行压频转换，然后将 8031 定时器 T0 设为定时状态，T1 设为计数状态，对脉冲信号进行计数。定时读取 T1 计数值，经“二 — 十”转换后送显示。本实验 8031 定时器 T0 为定时，T1 为计数，方式字 51H。

4. 实验器材:

（1）G2010+实验平台　1　台；　（2）G6W 仿真器　1　台；
（3）连线　若干　根；　（4）计算机　1　台

5. 接线方案：



6. 实验步骤：

（1）把“模拟信号发生器”的Vin0 孔连V/F转换电路LM331 的V2 孔，V/F转换电路LM331 的Fout孔连“分频器”的Fx孔，分频器的Fx/2 孔连“总线插孔”的P3.5(定时器T1)孔。

（2）设定工作模式为模式 2，即程序空间在仿真器上，数据空间在用户板上。

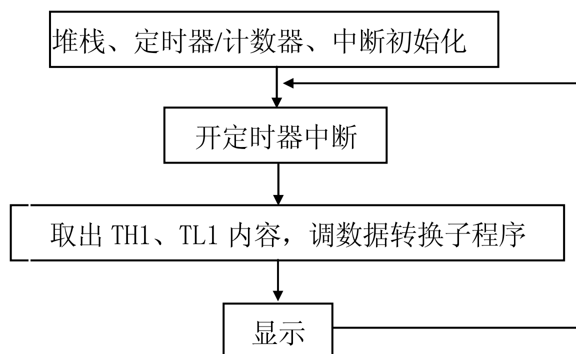
（3）硬件调试：

SICElab - G2010+实验平台加电以后，用示波器在 V/F 转换电路的 Fout 孔 即可观察到一脉冲波形，转动“模拟信号发生器”的电位器，输出脉冲频率会发生变化。

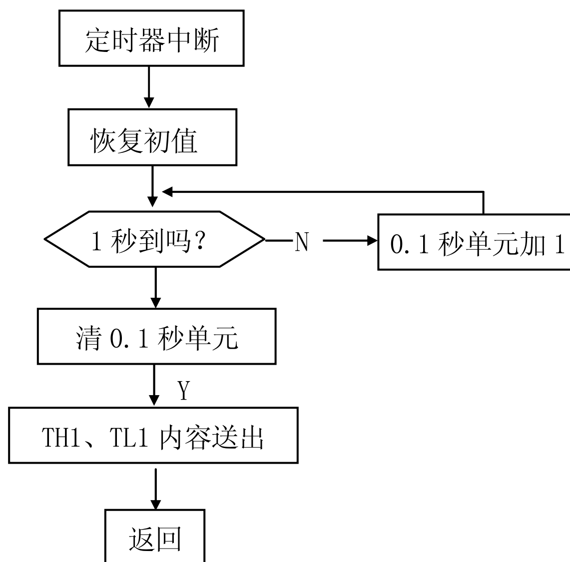
（4）编辑程序、编译程序。首先将断点设在中断服务程序入口地址上，运行程序 ，如果响应断点，则表明中断初始化程序正确，如果碰不到断点则应检查本实验初始化程序部分软件是否有错。将断点设在中断服务程序体中，检查 T1 计数是否与输入信号作相

应变化。再调试二进制翻十进制子程序。调试程序，排除软件错误，观察 6 位显示器显示数字与输入信号是否对应变化，不断修改程序，直至达到设计要求。

7．程序框图：



主程序流程图



中断处理子程序

8．程序清单：（B04.ASM）

实验五 力测量实验

1.实验目的:

了解力-电信号转换的基本工作原理，掌握ADC0809的使用方法，提高数据处理程序的设计和调试能力。

2.实验内容:

编写并调试出一个实验程序，其功能将一力施加于压力传感器金属弹性元件表面，SICElab-G2010+上数码管显示力的数据，并随力的大小而变化。

3．实验器材:

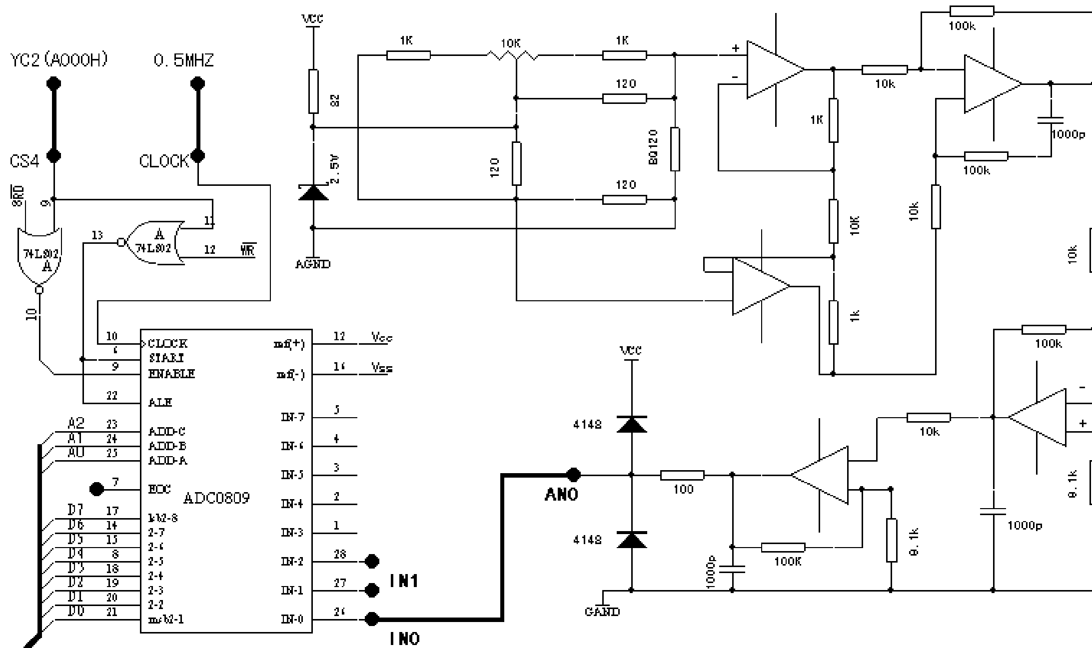
（1）G2010+实验平台　　1　　台

（2）G6W 仿真器　　1　　台

（3）连线　　若干　根

（4）计算机　　1　　台

4．实验线路:

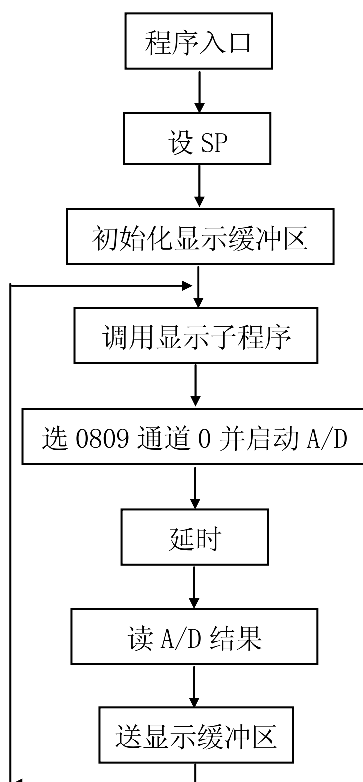


① 当在应变片上施加一力时,引起电桥不平衡,压力信号转换为微弱的电压信号,经LM324运算放大器，把信号放大至 0 - 5V，作为 ADC0809 输入信号。

② ADC0809 能与 CPU 直接接口，其输入电压为 0 – 5V，本实验中以 A2.A1.A0 作为通道地址线，CPU 对 0809 执行写操作时锁存通道地址。

③ 从实验原理图可以看出“译码器”的 YC2 作为 0809 片选信号,所以 0809 地址为:0A000H

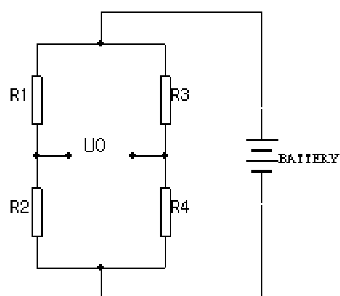
5．程序框图:



6. 工作原理:

将金属丝电阻应变片粘附在弹簧片的表面，弹簧片在力的作用下发生形变，而电阻应变片也随着弹簧片一起变形，这将导至电阻应变片电阻的变化。弹簧片受的力越大，形变也越大，电阻应变片电阻的变化也越大，测量出电阻应变片电阻的变化，就可以计算出弹簧片受力的大小。

图为应变片电桥测量电路，由应变片的电阻R1 和另外三个电阻R2、R3、R4 构成桥路，当电桥平衡时（即电阻应变片未受力作用时），R1=R2=R3=R4=R，此时电桥的输出U_0=0，当应变受力后，R1 发生变化，使R1、R3≠R2、R4，电桥输出U_0≠0，并有： $U_0 \approx \pm\frac{1}{4}\cdot\frac{\Delta R}{R}U \approx \pm\frac{K_0\varepsilon}{4}U$



应变片电桥测量电路

7. 实验步骤:

（1）设定工作模式为模式 2，即程序空间在仿真器上，数据空间在用户板上。

（2）“译码器”的 YC2 孔连数模转换 AD0809 的 CS4 孔，“脉冲源”的 0.5MHZ 孔连 AD0809 的 CLOCK 孔，IN0 孔（AD0809 的 0 通道）连 AN0 孔（压力传感器的输出孔）。

（3）硬件调试：在弹性元件表面施加一力。

（3．0） G2010+G6W 连 PC 机，在 WINDOWS 调试环境下点击外部数据窗口图标（ ），在打开的 XDATA 窗口中的 0A000H 地址上写入 00H（即选择 AD0809 的 0 通道），以启动 AD0809 对 IN0 上输入电压进行 A/D 转换，然后点击刷新图标（ ），读出 AD 转换结果。

（3．1） G2010+G6W 连 PC 机，在 DOS 调试环境下，把光标移入 XDATA 窗中，按 Alt+G，在打开的“Goto Address”窗口中输入“0A000H”，再按“回车”键，在 0A000H 地址上打入“00H”以启动 AD0809 对 IN0 上输入电压进行 A/D 转换，然后屏幕上显示的值即是读出的 AD 转换结果。

（4）输入程序，编译。在读取 AD 转换指令后设置断点，在弹性元件施加一力，全速运行，如果碰到断点，再检查读出 A/D 转换结果，数据是否与 Vin0 相对应，否则应查程序或硬件。再全速运行程序，修改程序错误使 SICElab-G2010+显示值随力的大小而变化，直至达到本实验的要求。

（5）可通过“压力传感器”框中的电位器，对电桥进行零点平衡调节。

8．程序清单：（B05.ASM）

实验六 直流电机转速测量与控制实验

1.实验目的:

了解霍尔器件工作原理及转速测量与控制的基本原理、基本方法，掌握DAC0832电路的接口技术和应用方法，提高实时控制系统的设计和调试能力。

2.实验内容:

设计并调试一个程序其功能为测量电机的转速，并在SICElab-G2010+实验平台显示器上显示出来，采用比例调节器方法，使电机转速稳定在某一设定值。此设定值可由G2010+实验平台上键盘输入。

3. 工作原理:

转速是工程上一个常用参数。旋转体的转速常以每秒钟或每分钟转数来表示，因此其单位为转/秒、转/分，也有时用角速度表示瞬时转速，这时的单位相应为孤度/秒。

转速的测量方法很多，由于转速是以单位时间内转数来衡量，在变换过程中多数是有规律的重复运动。霍尔开关传感器正由于其体积小，无触点，动态特性好，使用寿命长等特点，故在测量转动物体旋转速度领域得到了广泛应用。

霍尔器件是由半导体材料制成的一种薄片，在垂直于平面方向上施加外磁场 B，在沿平面方向两端加外电场，则使电子在磁场中运动，结果在器件的两个侧面之间产生霍尔电势。其大小和外磁场及电流大小成比例。

本实验选用美国史普拉格公司（SPRAGUE）生产的 3000 系列霍尔开关传感器 3013T，它是一种硅单片集成电路，器件的内部含有稳压电路、霍尔电势发生器、放大器、史密特触发器和集电极开路输出电路，具有工作电压范围宽、可靠性高、外电路简单、输出电平可与各种数字电路兼容等特点。器件采用三端平塑封装。引出端功能符号如下：

引出端序号	1	2	3
功能	电源	地	输出
符号	VC1	GND	OUT

我们根据霍尔效应原理，将一块永久磁钢固定在电机转轴上的转盘边沿，转盘随测轴旋转，磁钢也将跟着同步旋转，在转盘附近安装一个霍尔器件 3013T，转盘随轴旋转时，受磁钢所产生的磁场的影响，霍尔器件输出脉冲信号，其频率和转速正比，测出脉冲的周期或频率即可计算出转速。

直流电机的转速与施加工于电机两端的电压大小有关。本实验用DAC0832控制输出到直流电机的电压，控制DAC0832的模拟输出信号量来控制电机的转速。当电机转速小于设定值时增大D/A输出电压，大于设定值时则减小D/A输出电压，从而使电机以某一速度恒速旋转。我们采用简单的比例调节器算法（简单的加一、减一法）。

比例调节器（P）的输出系统式为： $y=K_pe(t)$

式中：Y——调节器的输出

e(t)——调节器的输入，一般为偏差值

K_P——比例系数

从上式可以看出，调节器的输出Y与输入偏差值e(t)成正比。因此，只要偏差e(t)一

出现就产生与之成比例的调节作用，具有调节及时的特点，这是一种最基本的调节规律。比例调节作用的大小除了与偏差e(t)有关外，主要取决于比例系数K_p，比例调节系数愈大，调节作用越强，动态特性也越大。反之，比例系数越小，调节作用越弱。对于大多数的惯性环节，K_p太大时将会引起自激振荡。比例调节的主要缺点是存在静差，对于扰动的惯性环节，K_p太大时将会引起自激振荡。对于扰动较大，惯性也比较大的系统，若采用单纯的比例调节器就难于兼顾动态和静态特性，需采用调节规律比较复杂的PI（比例积分调节器）或PID（比例、积分、微分调节器）算法。

4. 实验器材:

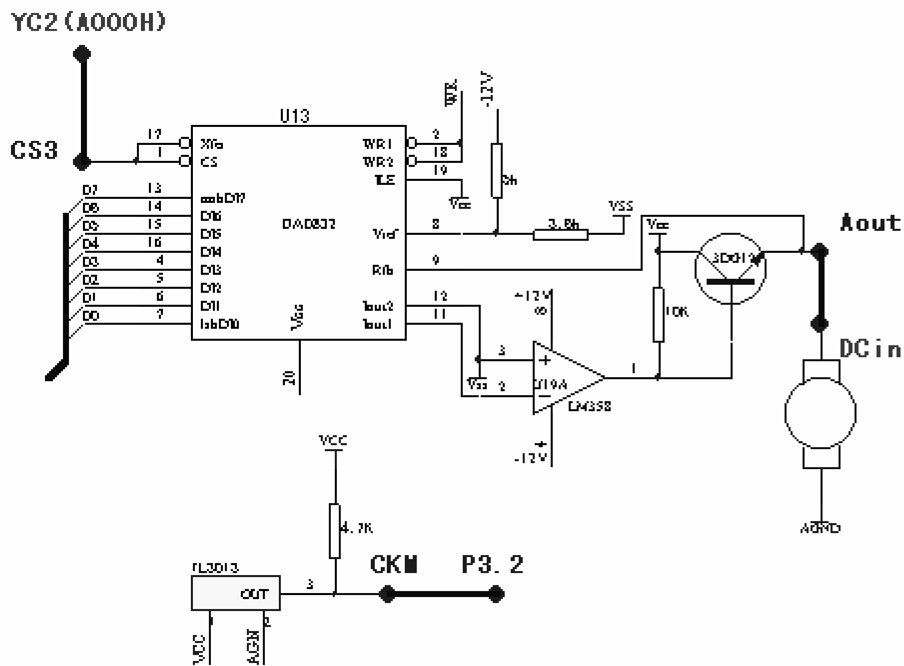
（1）G2010+实验平台　1　　台

（2）G6W 仿真器　　　1　　台

（3）连线　　　　　若干　　根

（4）计算机　　　　　1　　台

5. 实验线路:



定时器 T1，工作于外部事件计数方式，对转速脉冲计数；T0 工作于定时器方式，均工作于方式 1。“译码器”的 YC2 孔作为 DAC0832 的片选端，故 DAC0832 地址为 0A000H-0AFFFH。

6. 实验步骤:

（1）设定工作模式为模式 2，即程序空间在仿真器上，数据空间在用户板上。把数模转换 DAC0832 输出 Aout 孔连直流电机 Dcin 孔，数模转换 DAC0832 的 CS3 孔连“译码器”的 YC2 孔，CKM 孔（霍尔器件输出孔）连 P3.2 孔。

（2）硬件诊断:

（2．0） G2010+G6W 连 PC 机，在 WINDOWS 调试环境下点击外部数据窗口图标（ ），在打开的 XDATA 窗口中的 0A000H 地址（即 DA0832 的片选空间）上写入 FFH，则 Aout 孔输出应为 5V,直流电机快速旋转;将 00H 写入,则 Aout 孔输出应为 0V, 直流电机停止转动。

（2．1） G2010+G6W 连 PC 机，在 DOS 调试环境下，把光标移入 XDATA 窗中，按 Alt+G，在打开的“Goto Address”窗口中输入“0A000H”，再按“回车”键，在 0A000H 地址上打入“0FFH” 则 Aout 孔输出应为 5V，直流电机快速旋转;将 00H 写入，则 Aout 孔输出应为 0V,直流电机停止转动.

（2．2）如选购了 G2K 仿真板，则还可直接在 G2010+实验平台的键盘上进行硬件诊断：设定工作模式为模式 2，即数码管显示为“PI EE”，然后按“MON”键使数码管显示为“ ’ ”，输入“A000”，按“+”，再输入“FF”，则 Aout 孔输出应为 5V，直流电机快速旋转；将 00H 写入，则 Aout 孔输出为 0V，电机停止转动。

用示波器观察 CKM 孔测试点，当圆盘转动时，霍尔器件输出一系列脉冲。

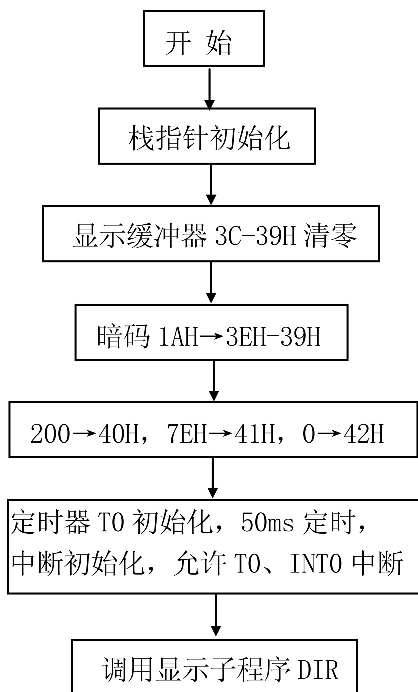
（3）编程并编译。首先将断点设在中断服务程序入口，运行程序 ，如果程序进入中断处理程序入口，则表明中断初始化程序正确，如果碰不到断点则首先应检查初始化程是否有错。把断点设在中断程序结束，检查在单位定时内，T1 计数值是否与电机转速符合。再调试二翻十子程序，最后调试整个实验程序，排除软件错误，连续运行时观察电机旋转工作状态与数码管上显示是否正确，修改程序直至达到本实验设计要求。注：本实验电机转速范围一般应为 35 - 50 转/分。

7.程序框图：

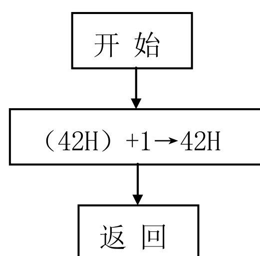
8．思考问题：

试编写一转速测量软件，测试电机转动周期 T，然后计算瞬时转速，并用 PID 调节使转速恒定在 25 转/分。

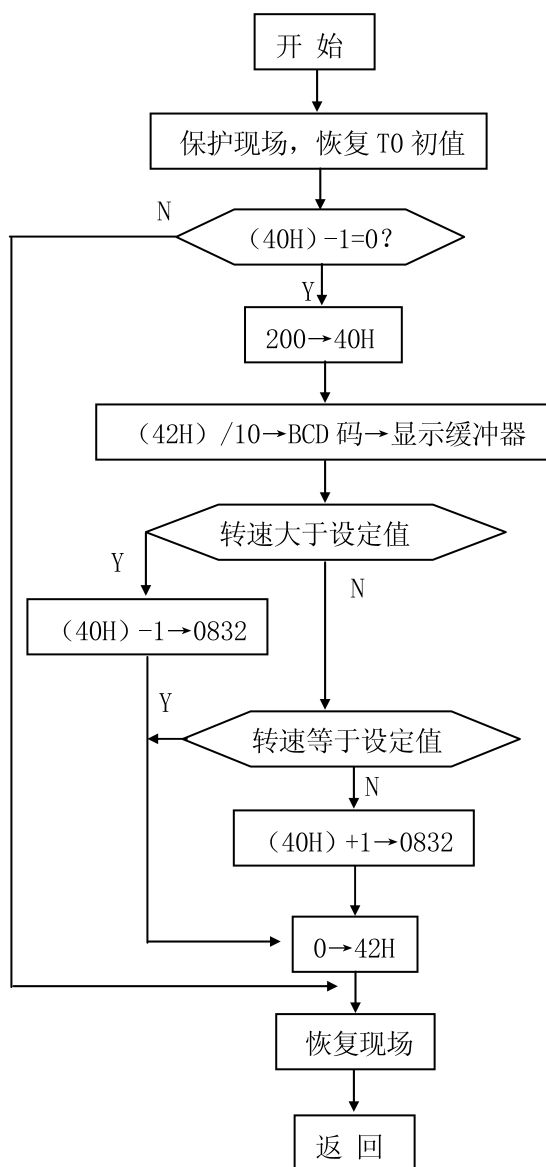
9．程序清单：（MCS51\B06.ASM）



转速测量与控制实验主程序框图



INT1 中断程序



T0 中断程序框图

实验七 点阵式LCD液晶显示屏实验

1. 实验目的：

学习获取字模的方法；学习 122X32A 液晶 LCD 的原理及编程方法；

2. 实验内容：

在 LCD 上显示“蓬勃远大”字样。

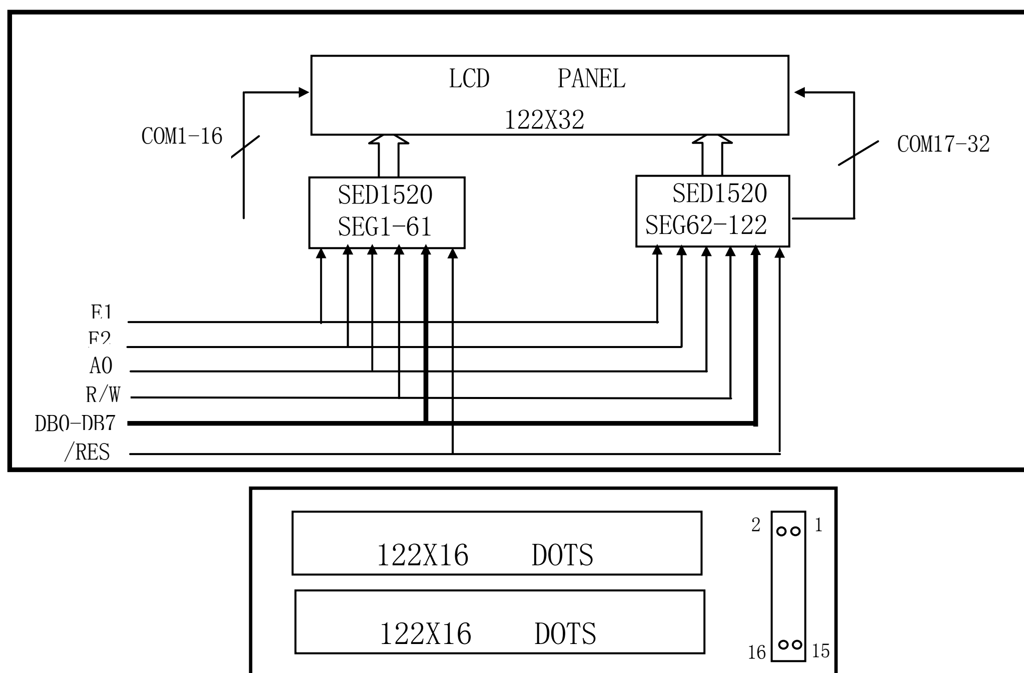
3. 实验器材：

（1）G2010+实验平台 1 台 （2）G6W 仿真器 1 台（3）连线 若干 根（4）计算机 1 台

4. 工作原理：

（1）我们选用的 122X32A 点阵式液晶 LCD，是内置 SED1520 液晶显示控制器的屏，它集行、列驱动器和控制器于一体，被广泛应用于小规模液晶显示模块中。SED1520 内置 2560 位显示 RAM 区，RAM 中的 1 位数据控制液晶屏上一个象素的亮、暗，“1”为亮，“0”为暗。它具有 16 个行驱动输出和 61 个列驱动输出，可以直接与 80 系列或 68 系列的 CPU 相连，驱动占空比为 1/32，显示内容：122X32 点。

（2）122X32A 点阵式液晶 LCD 电路图：



引　脚	符　号	电　平	说　明
1	VDD	5．0V	电源电压
2	GND	0V	接地（GND）
3	V0	可调	LCD 驱动电压（对比度调节）
4	/RES	H/L	复位信号
5	E1	H，H→L	片使能信号 1
6	E2	H，H→L	片使能信号 2
7	R/W	H/L	H：读；L：写
8	A0	H/L	H：数据；L：指令
9	DB0	H/L	数据位 0
10	DB1	H/L	数据位 1
11	DB2	H/L	数据位 2
12	DB3	H/L	数据位 3
13	DB4	H/L	数据位 4
14	DB5	H/L	数据位 5
15	DB6	H/L	数据位 6
16	DB7	H/L	数据位 7

（3）极限参数：

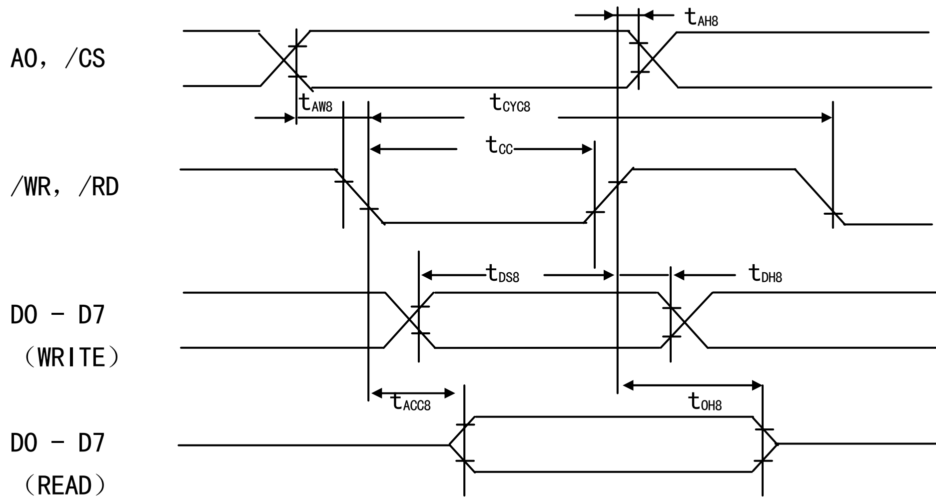
名称	符号	测试条件	标准值		单位
			最小值	最大值	
电源电压	V_{DD}-V_{CC}	Ta=25℃	0	6．5	V
LCD 驱动电压	V_{DD}-V_0		0	12．0	V
输入电压	V_i		0	V_{DD}	V

（4）电参数：

名称		符号	测试条件	标准值			单位
				最小值	典型值	最大值	
电压	逻辑	V_{DD}-V_{SS}	-	4．75	5．0	5．25	V
	LCD	V_{DD}-V_0	-	4．5	5．5	6．5	V
电流	逻辑	I_{DD}	-	-	2．5	-	mA
	LCD	I_{EE}	-	-	2．0	-	mA
LCD 工作电压（推荐值）		V_{DD}-V_0	0℃	-	6．2	-	V
			25℃	-	5．5	-	V
			40℃	-	4．8	-	V
输入电压	高电平	V_{IH}	高电平	0．7V_{DD}	-	V_{DD}	V
	低电平	V_{IL}	低电平	0	-	0．3V_{DD}	V

Read/Write Timing for the 80-port MPU(Ta=0 to 75℃,V_{SS}=.0V 0%)

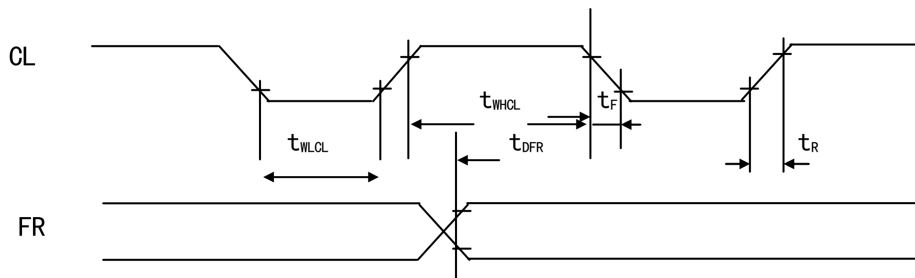
(5)



Parameter	Signal	Symbol	Condition	Rating			Unit
				Min	Type	Max	
Address hold time	A0, /CS	t_{AH8}		10	-	-	ns
Address set-up time		t_{AW8}		20	-	-	ns
System cycle time	/WR, /RD	t_{cyc8}		1000	-	--	ns
Control pulse width		tcc		200	-	-	ns
Data Set-up time	D0-D7	t_{DS8}		80	-	-	ns
Data hold time		t_{DH8}		10	-	-	ns
RD access time		t_{ACC8}	CL=100PF	-	-	90	ns
Output disable time		t_{OH8}		10	-	60	ns

★The ratings when V_{SS}=.0V are approximately 100% higher then when V_{SS}=.0V.

(6) Control timimg for 80 - port/68 - port display(Ta = 0 to 75℃, V_{SS} = .0V 0 %)



Parameter	Signal	Symbol	Condition	Rating			Unit
				Min	Type	Max	
LOW pulse width	CL	t_{WLCL}		35	–	–	ns
HIGH pulse width		t_{WHCL}		35	–	–	ns
Rising time		t_R		–	30	150	ns
Falling time		t_F		–	30	150	ns
FR delay time	FR	t_{DFR}	Input timing	-2.0	0.2	2.0	ns
			Ouput timing Cl=100pf	–	0.2	0.4	ns

(7)指令功能：SED1520 有 13 条指令，从作用上可分为两大类：一类为显示方式的设置指令，前 6 条指令为这一类指令，它们只需在初始化程序中写入一次即可。另一类为显示数据读/写操作的指令，从第 7 条往下（包括读状态字）都是这类指令。

Command		RD	WR	A0	D7	D6	D5	D4	D3	D2	D1	D0
1	Display ON/OFF	1	0	0	1	0	1	0	1	1	1	0/1
2	Display START Line	1	0	0	1	1	0	Display START address(0-31)				
3	Page Address Set	1	0	0	1	0	1	1	1	0	Page(0-3)	
4	Column(Segment) Address Set	1	0	0	0	Column address (0-79)						
5	Status Read	0	1	0	BUSY	ACC	ON/OFF	RET	0	0	0	0
6	Write Display Data	1	0	1	Write Data							
7	Read Display Data	0	1	1	Read Data							
8	ADC Selet	1	0	0	1	0	1	0	0	0	0	0/1
9	Static Drive ON/OFF	1	0	0	1	0	1	0	0	1	0	0/1
10	Duty Selet	1	0	0	1	0	1	0	1	0	0	0/1
11	Read Modify Write	1	0	0	1	1	1	0	0	0	0	0
12	End	1	0	0	1	1	1	0	1	1	1	0
13	Reset	1	0	0	1	1	1	0	0	0	1	0

COM1: Switches the entire display ON or OFF,regardless of the Display RAM data or the internal status★★

COM2: Determines the line of RAM data to be displayed at the display top line

COM3: Sets the page of the Display RAM in the page address register

COM4: Sets the column address of the Display RAM in the column address register

COM5: Reads the status

BUSY	1: Busy(intemal processing)	0: READY status
ADC	1: Rightward(forward)output	0: leftward(reverse)output
ON/OFF	1: Display OFF	0: Display ON
RET	1: Resetting	0: Normal

COM6: Writes the data on the data bus to RAM

COM7: Reads data from the Display RAM onto the data bus

COM8: Used to reverse the correspondence between the Display RAM column address and segment driver output ports

0: Rightward(forward)output 1: leftward(reverse)output

COM9: Selets normal display operation or static all-lit drive display operation.

1: Static drive(Power Save)★★ 0: Normal display operation

COM10: Selects the duty factor for driving LCD cells

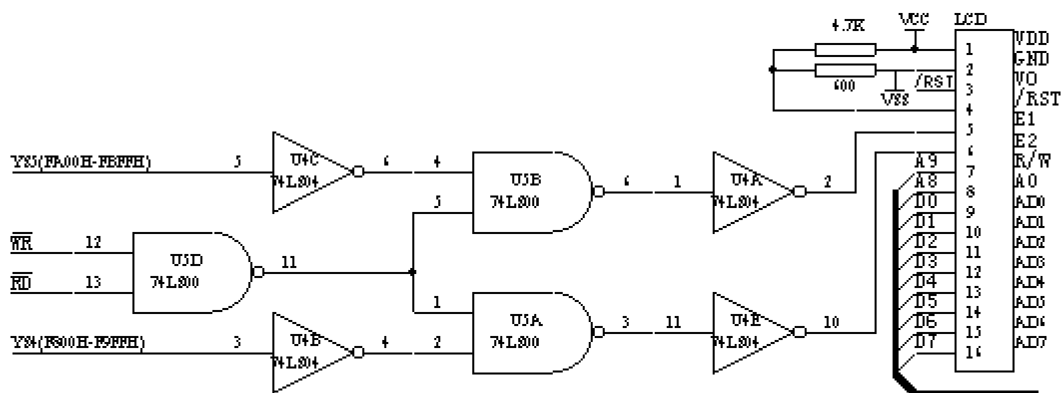
1: 1/32 duty 0: 1/16 duty

COM11: Increments the column address counter by one only when display data written but not when it is read.

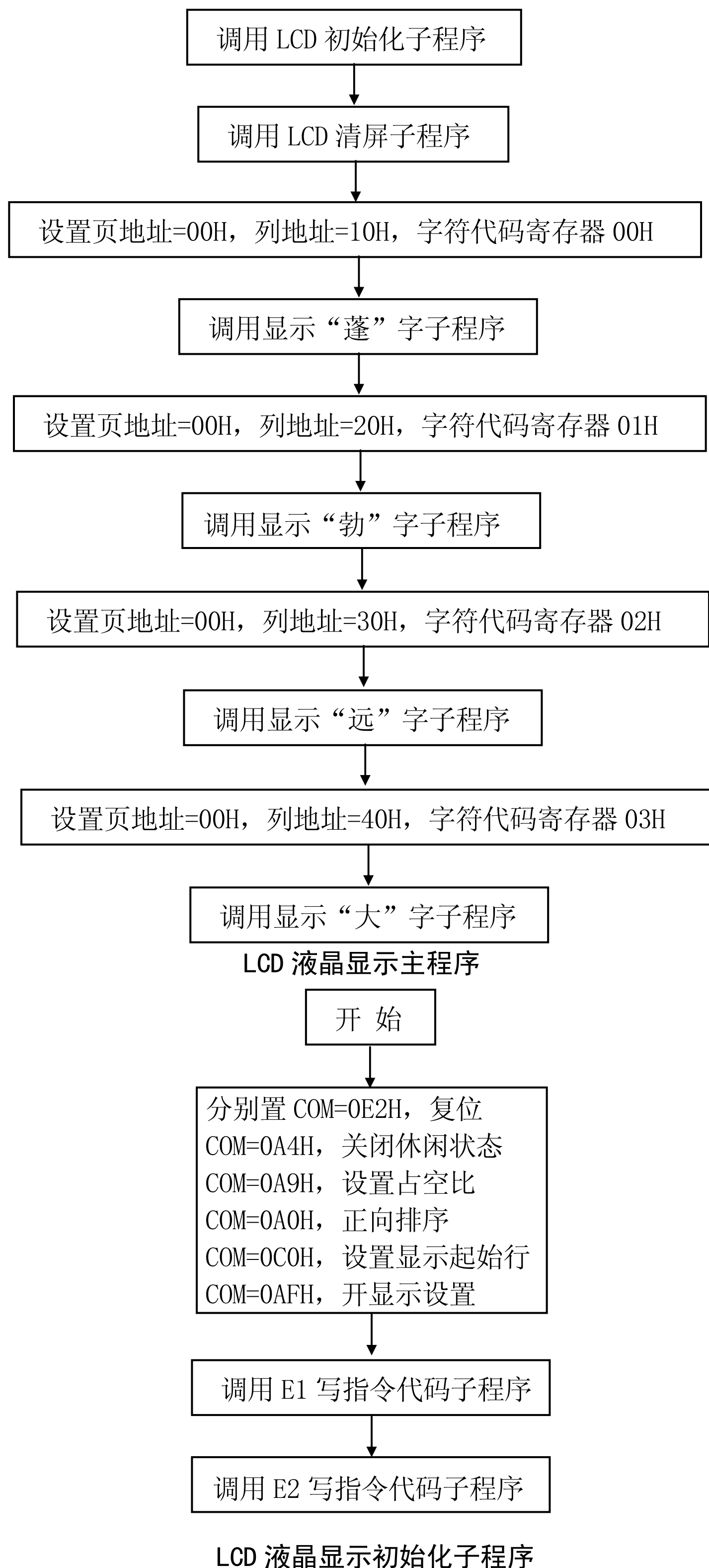
COM12: Cancels the Ready Modify Write Mode

COM13: Resets the Display START line to the 1st line in the register.Resets the column address counter to 0 and page addess register to 3

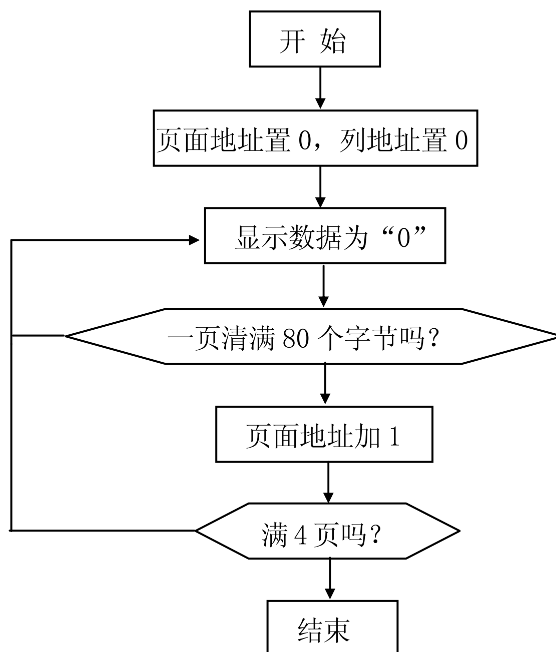
5.接线方案：



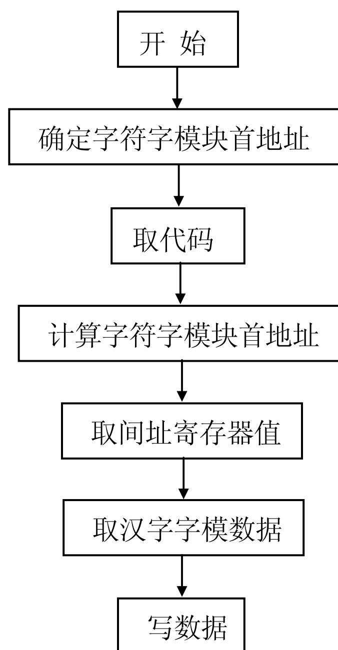
6.程序框图：



LCD 液晶显示初始化子程序



LCD 液晶显示清屏子程序



LCD 显示子程序

7．实验步骤：

（1）设定工作模式为模式 2，即程序空间在仿真器上，数据空间在用户板上。

（2）系统分配给 LCD 的地址为：0FA00H-0FBFFH，0F800H-0F9FFH

（3）设计称序并执行，观察结果。

8．思考问题：

（1）显示“赛思仪器”字样　　　　　　　　　　（2）修改程序，使其上、下、左、右移动。

9．程序清单：（MCS51\B07.ASM）

实验八　温度测量实验

1、实验目的：

了解热敏电阻测温基本工作原理及小信号放大器工作原理和零点、增益的调整方法。

2、实验内容：

使用电桥将热敏电阻阻值变化转换为电压信号放大以后经 A/D 转换为数字量由 CPU 处理，在 LED 上显示出来。

3、工作原理：

温度测量采用热敏元件作为传感器，常用的温度传感器有热敏电阻、热电偶、集成温度传感器等。其中热敏电阻为价格低使用方便。根据电阻和温度关系有负温度系数，正温度系数和临界温度热敏电阻。

4、实验步骤：

（1）系统连接：

把“温度传感器”框中的 AN2 孔连 AD0809 的 0 通道 IN0；“脉冲源”框中的 0.5MHZ 孔连 AD0809 的 CLOOK 孔；“译码器”的 YC2（0A000H）孔连 AD0809 的 CS4 孔。

（2）设定工作模式为模式 2，即程序空间在仿真器上，数据空间在用户板上。

（3）硬件调试：在热敏电阻表面加温。

（3．0） G2010+G6W 连 PC 机，在 WINDOWS 调试环境下点击外部数据窗口图标（），在打开的 XDATA 窗口中的 0A000H 地址上写入 00H（即选择 AD0809 的 0 通道），以启动 0809 对 IN0 上输入电压进行 A/D 转换，然后点击刷新图标（），读出 AD 转换结果。

（3．1） G2010+G6W 连 PC 机，在 DOS 调试环境下，把光标移入 XDATA 窗中，按 Alt+G，在打开的“Goto　Address”窗口中输入“0A000H”，再按“回车”键，在 0A000H 地址上打入“00H”以启动 0809 对 IN0 上输入电压进行 A/D 转换，然后屏幕上显示的值即是读出的 AD 转换结果。

（4）输入程序，编译。全速运行程序，修改程序错误使 SICElab-G2010 显示值随 W 温度的高低而变化，直至达到本实验的要求。

（5）可通过“温度传感器”框中的电位器，对电桥进行零点平衡调节

5、接线方案：

图中使用的热敏电阻为负温度系数热敏电阻，温度愈高，电阻愈小，运放输出的电压愈高。该电压信号输入到 AD0809 的第 0 号通道进行转换。读 AD0809 即可得到环境温度值。

6、程序框图：

设 SP

初始化显示缓冲区

调用显示子程序

选 0809 通道 0 并启动 A/D

延时

送显示缓冲区

读 A/D 结果

7、思考问题：

请把 16 进制数转换成十进制数在数码管上显示出来。

8、程序清单：（MCS51\B08.ASM）

实验九　　微型打印机打印字符

1. 实验目的：

了解微型打印机与 CPU 的连接方法，掌握编程技巧。

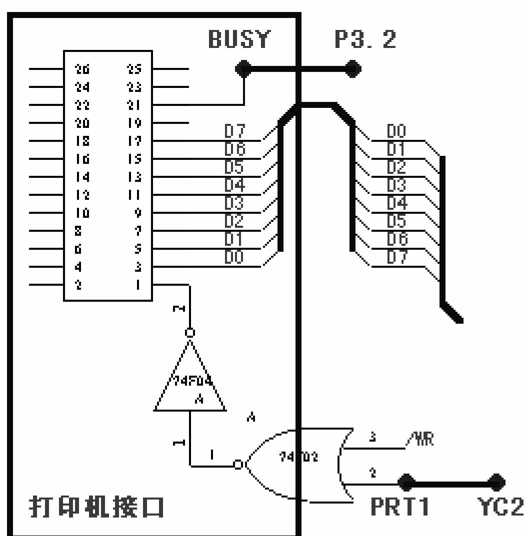
2. 实验内容：

使用微型打印机打印 RAM 区中的 ASCII 码字符。

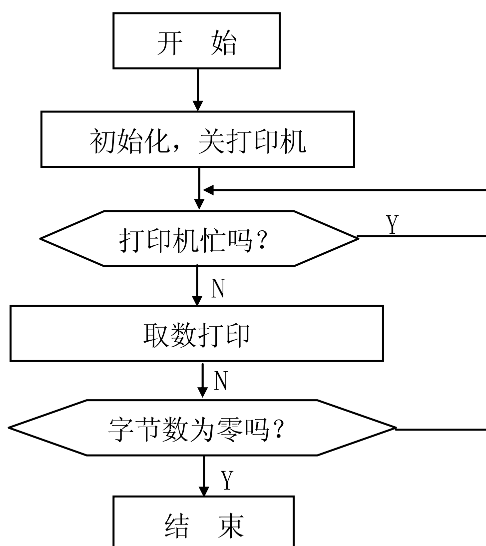
3. 实验器材：

（1）G2010+实验平台 1 台（2）连线 若干（3）微型打印机 1 台 （4）G6W 仿真器 1 台

4. 接线方案：



5. 程序框图：



6. 实验步骤：

把微型打印机与打印机接口用扁平电缆相连，PRT1 孔连“译码器”的 YC2 孔，“总线插孔”区的 P3.2 孔连 BUSY 孔，设计并执行程序。

7、思考问题：

打印汉字“赛思仪器”

8.程序清单：（MCS51\B09.ASM）

实验十　点阵 LED 实验

1. 实验目的：

掌握点阵 LED 的原理和程序设计方法；掌握 74LS164 扩展并口的方法。

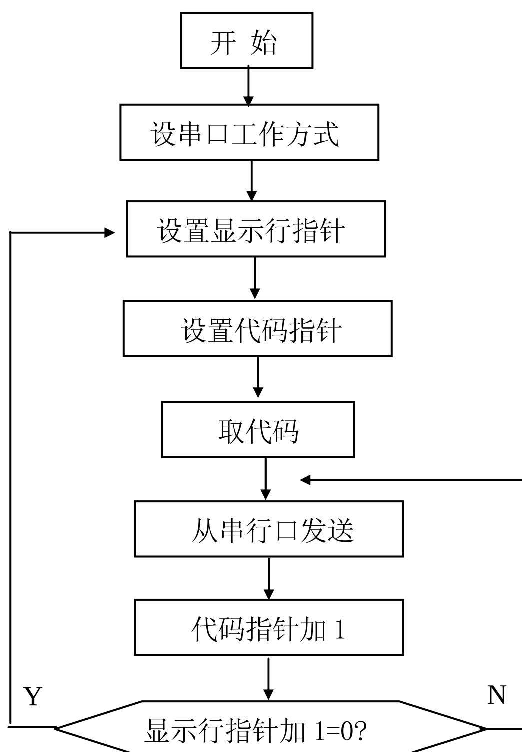
2. 实验内容：

在点阵 LED 上显示”5”

3. 实验器材：

（1）2010+实验平台　1　台　（2）连线　若干　根　（3）G6W 仿真器　1　台

4. 程序框图：



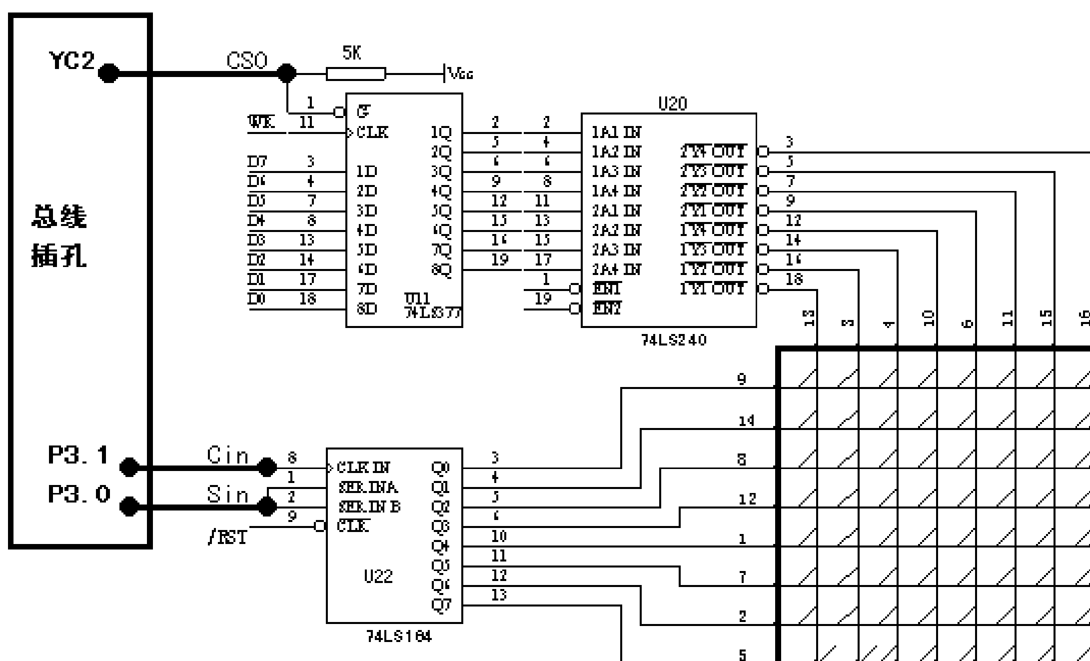
5. 实验步骤：

（1）设定工作模式为模式 2，即程序空间在仿真器上，数据空间在用户板上。

（2）“译码器”YC2（0A000H）孔连“点阵 LED”左侧 CS0 孔,“总线插孔”中 P3.0 孔连 Sin 孔,P3.1 孔连 Cin 孔.

（3）设计程序,调试并通过.

6. 接线方案：



7.程序清单:(MCS51/B10.ASM)

实验十一　红外线遥控实验

1.实验目的：

（1）了解红外遥控电路的原理，及编码方法。

（2）了解远程控制的一般原理和方法。

（3）学习如何编写红外发射和接收程序。

（4）了解单片机控制外部设备的常用电路。

2.实验内容：

利用 G2010+实验平台上的红外线接收、发送器件，编写程序发送和接红外信号，实现近距离的无线通信。

3.实验器材：

（1）G2010+实验平台　2　台　（2）连线　　若干　　根　（3）G6W 仿真器　2 台

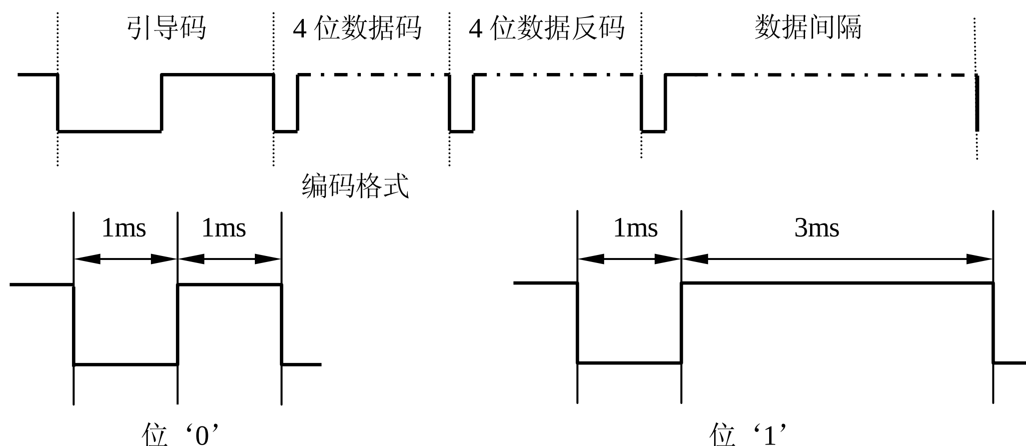
4. 实验说明：

红外遥控为现在最常用的近距离无线通信方式，它是将数字信号用红外线发送出去。为了能让受控设备能识别信号，要将数字信号编码，现今红外有很多编码标准，常见有 PHILIPS 的 RC5 格式和 NEC 格式。本实验为了简化，采用我们自己设计的一种编码方式。下面将详细说明。

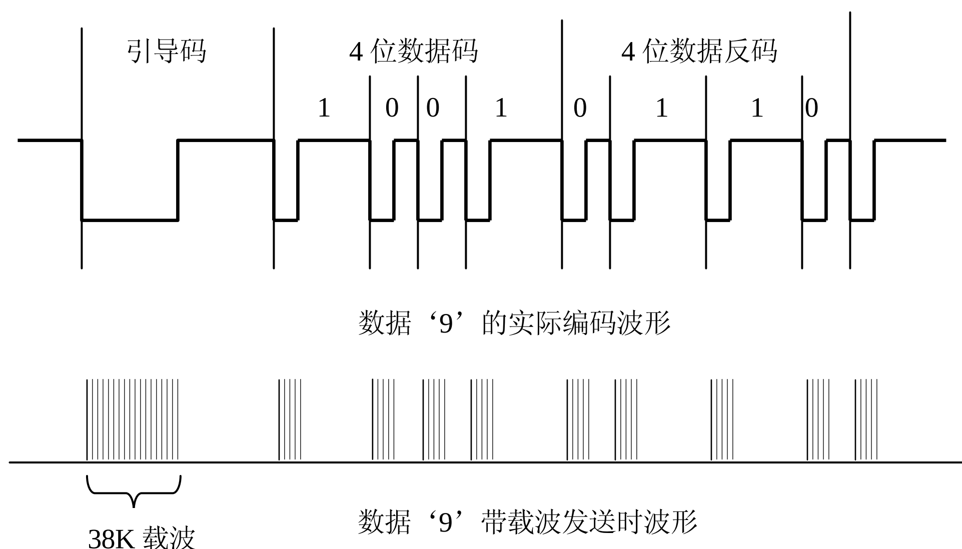
在普通场合，有很多红外发射源，有白炽灯、日光、发热体，这些都会干扰红外信号，所以在发射时，还要数字将脉冲信号调制在 30K-40K 的载波上，以抑制这些红外干扰。本实验采用最常用的 38K 载波。为了抗干扰，还可以在接收处适当地加一些隔离。本实验接收部分采用的是一体化接收头，将信号解调和放大全部做在一起，提高了可靠性。这样，接收头送到单片机的就是编码的数据信号，而不是调制信号，数据的解码通过单片机来完成。

本实验使用的编码包括四部分，引导码，4 位数据码，4 位数据反码和数据间隔。引导码用于标识一个数据的开始，数据码为有效数据，数据反码是将有效数据取反后编码，用于提高数据的识别率。

引导码由 5ms 低电平和 5ms 的高电平组成，数字位‘0’由 1ms 低电平和 1ms 高电平组成，数字位‘1’由 1ms 电平和 3ms 高电平组成。数据间隔为 20ms。



编码格式



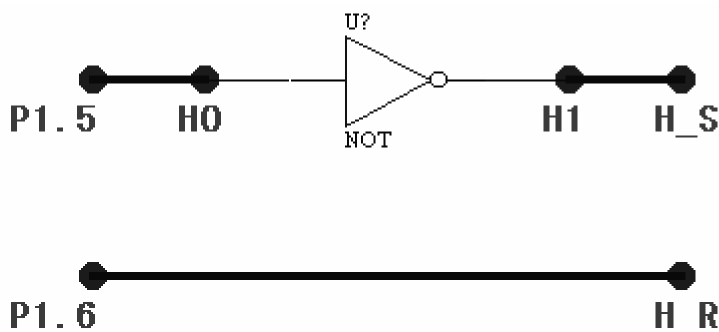
数据‘9’的实际编码波形

数据‘9’带载波发送时波形

在用脉冲控制红外发射管时，是低有效。即当输出低电平时，红外管导通发光。单片机输出的脉冲信号被反向驱动后，驱动红外管产生脉冲信号。

接收红外编码信号时，判断信号变化时间长短，就可以对信号进行译码，得到对方发过来的数据。

5. 接线方案：



6. 实验步骤：

（1）本实验需要两台实验系统 ：一台发送,另一台接收。

（2）设定工作模式为模式 2，即程序空间在仿真器上，数据空间在用户板上。

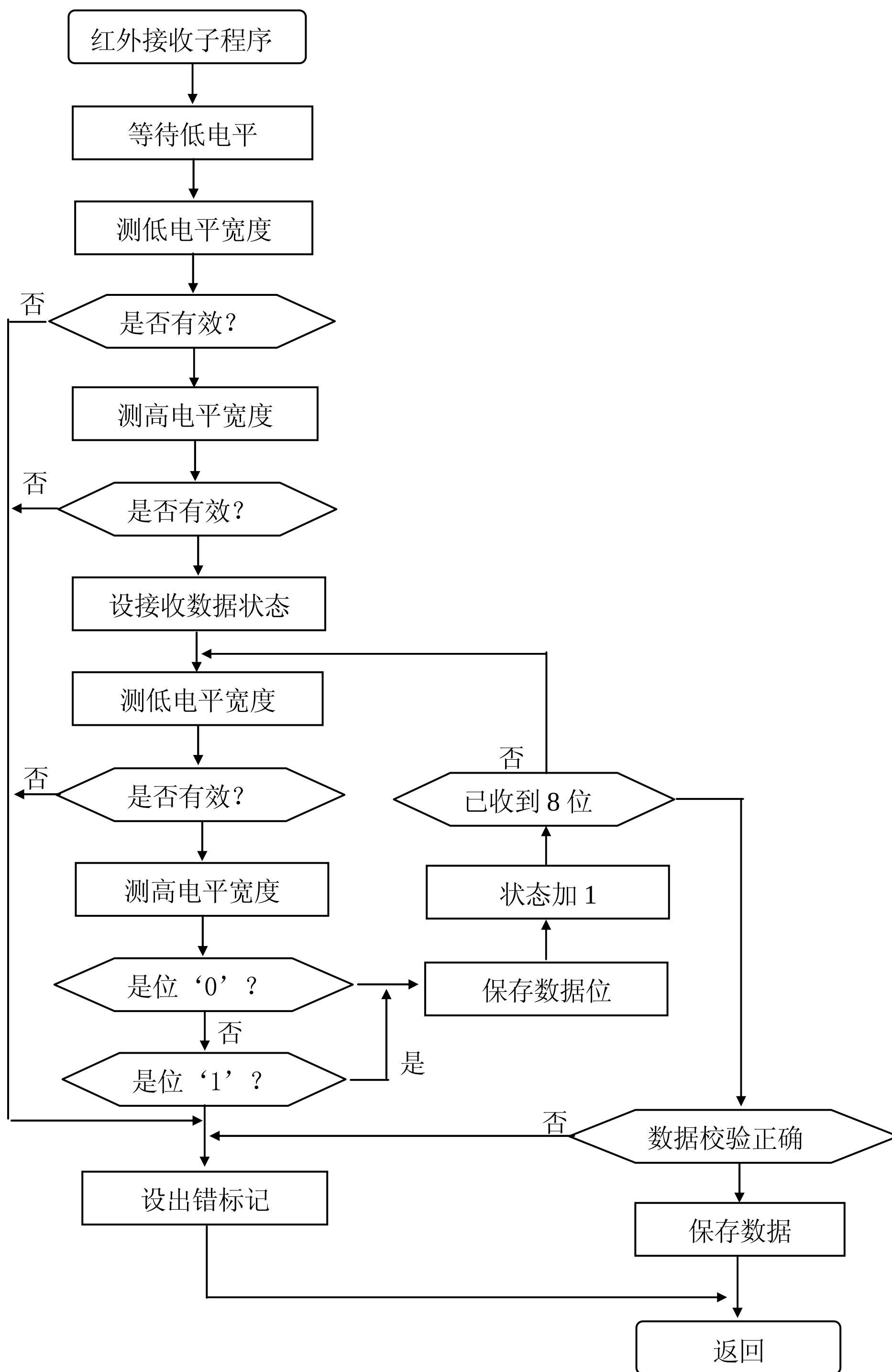
（3）P1.5 孔经反向器后接 H_S 孔 ， P1.6 孔接 H_R 孔。

（4）设计程序,调试并通过.

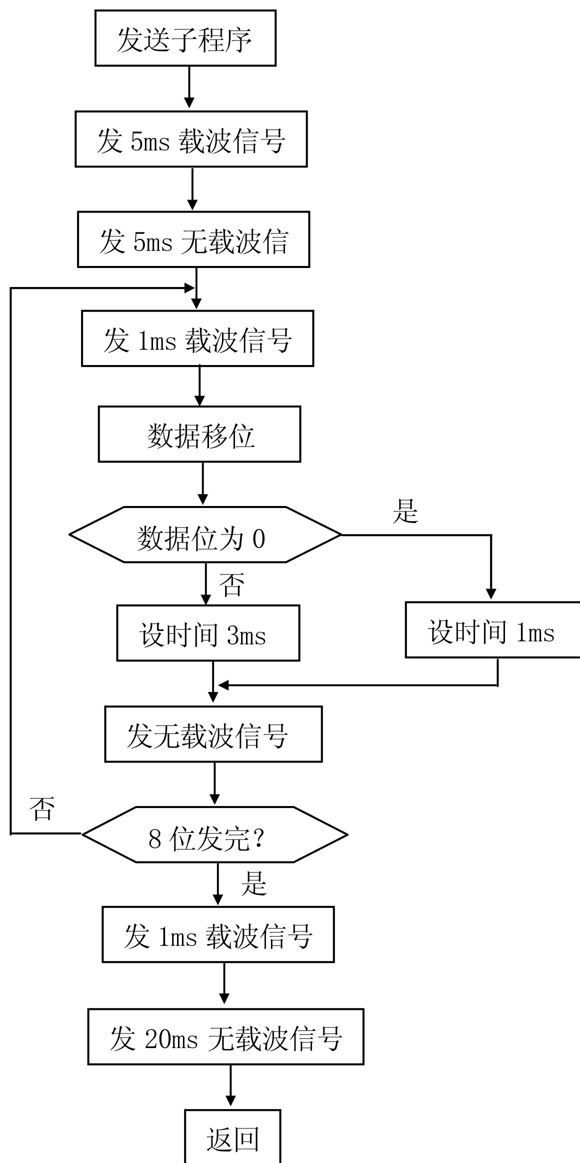
7. 程序清单:(MCS51/B11.ASM)

8．程序框图：

这里只给出红外发送和接收的子程序框图，有关键盘和显示的程序框图请参见相关部分。



红外接收子程序框图


发送子程序
发 5ms 载波信号
发 5ms 无载波信
发 1ms 载波信号
数据移位
数据位为 0
是
否
设时间 3ms
设时间 1ms
发无载波信号
否
8 位发完?
是
发 1ms 载波信号
发 20ms 无载波信号
返回

红外发射子程序框图

第❸节　MCS51“探索式”实验例程

实验一　最小系统组成实验（AT89C51）

1．实验目的：

通过最小系统组成实验，掌握单片机系统的时钟电路、复位电路工作原理。

2．实验器材：

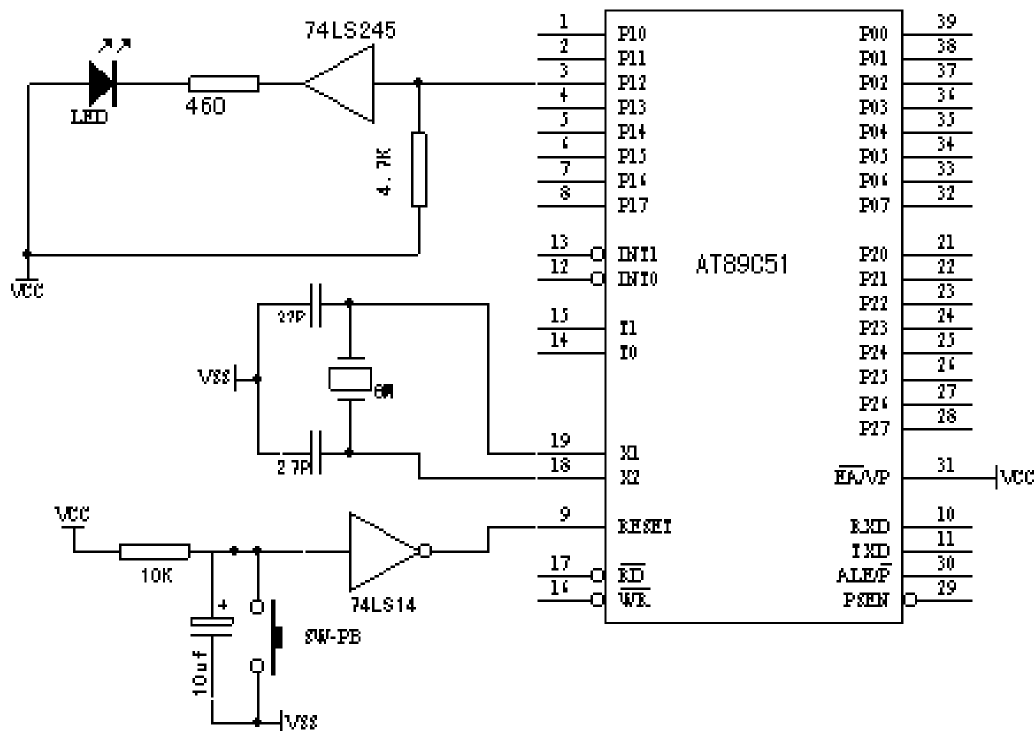
（1）G2010+实验平台 1 台　　（2）G6W 仿真器　1 台
（3）连线　若干 根　　（4）编程器 1 台
（5）AT89C51 芯片 1 片　　（6）万用表 1 块
（7）示波器 1 台　　(8) POD51 仿真头

3．实验原理：

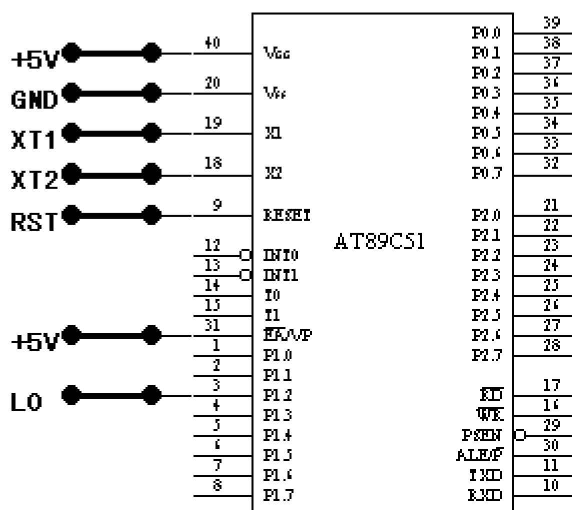
（1）MCS—51 单片机内部有一个用于构成振荡器的高增益反向放大器，引脚 XTAL1 和 XTAL2 分别是此放大器的输入和输出端，其时钟可由内部和外部两种方式产生。由于采用内部方式时，电路简单，实际使用中常采用这种方式，如图中所示，外接晶体以及电容 C1 和 C2 构成并联谐振电路，接在放大器的反馈回路中（XTAL1 和 XTAL2），晶体振荡器在 2-12M 之间，C1、C2 的值为 20P 左右。G2010+实验平台上已把晶体与电容连接好，并以插孔方式引出。(总线插孔区）

（2）复位的目的是初始化单片机内部的某些特殊功能寄存器。单片机的复位是靠外电路实现的，在正常运行情况下，只要 RST 引脚出现两个机器周期以上时间的高电平，即可引起系统复位，复位后单片机内部一些特殊功能寄存器（SFR）被设置成一定的值，如 PC=0000H，SP=07H 等。复位操作有两种情况，即上电复位和手动（开关）复位。由于干扰等因素造成程序不能正常运行时，就需要开关复位。下图中由 R1、C1 和 74LS14 组成上电复位电路，由按扭 SW-PB 和 74LS14 组成手动复位电路。上电时，由于电容两端的电压不能突变，使得 74LS14 输入端的电平为低电平，其输出端为高电平，随着电源对电容 C1 的充电，在某一时刻，当电容两端的电压高于输入低电平门限时，74LS14 输出变成低电平，使单片机脱离复位状态进入工作状态。R 取值在 10K 左右，C 取值为 10μF;在正常运行过程中，电容两端的电压接近 5V，74LS14 输出为低电平，当按下按扭 SW-PB 时，电容就会通过按扭对地短路放电，使 74LS14 输入快速变为低电平，输出变为高电平，当松开接扭时，过程与上电复位时一样。现在在实际应用中，一般多使用专用的复位芯片如 upA7705。G2010 实验平台上用了 upA7705 专用复位电路，并以 RST 和/RST 插孔引出。AT89C51 单片机/EA（引脚 31）必须接高电平，如图。

4．接线方案：

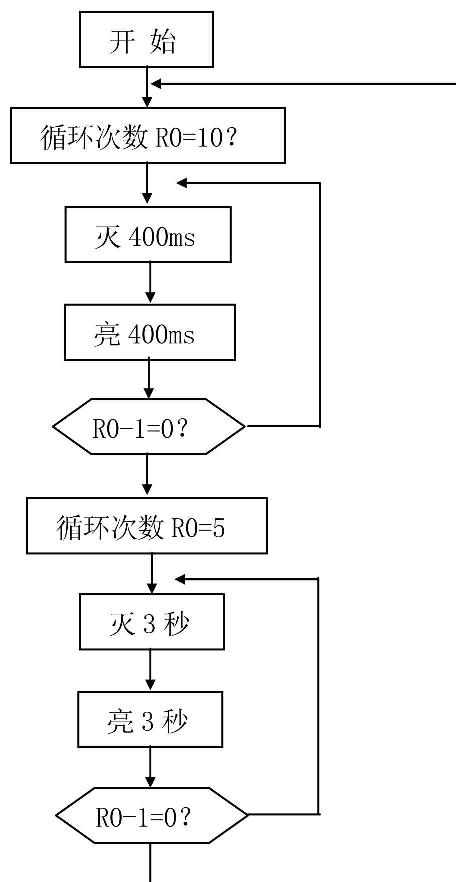


最小系统组成实验原理图



最小系统组成接线方案

5．程序框图：



6．实验步骤：

（1）把 Lab51CPU 板取下。把 POD51 仿真头与 G6W 仿真器连接好，而后把 POD51 仿真头插入 G2010+实验平台的锁紧插座中。按上图连接好硬件。

（2）按图示中的接线方案，把实验电路搭试好。

（3）硬件检测：在调试环境的 CPU 窗口中，直接点击 P1.2 位，置 1 时发光二极管亮，反之，发光二极管灭。否则，硬件有故障。

（4）编写程序，要求开始时发光二管灭 400mS，亮 400mS，10 次以后灭 3 秒，亮 3 秒，进行 5 次，然后重复上述过程。

（5）利用仿真器把程序调试好，用编程器把代码烧录在 AT89C51 芯片中。

（6）把 POD51 仿真头拔出，把 AT89C51 芯片插入锁进插座中，加电运行程序，在后一阶段按复位按扭，观察程序的运行状态。

（7）用示波器观察 AT89C51 的第 18 引脚 XTAL2，即 G2010+实验平台上 XT2 孔的波形。

（8）用万用表测量 AT89C51 的第 9 引脚 RST，即 G2010+实验平台上 RST 孔的电压，观察当按下和松开复位按扭[RST]键时，其电平的变化。

7．程序清单：（MCS51\C01.ASM）

实验二 程序存储器扩展实验

1. 实验目的：

（1）掌握单片机系统程序存储器的扩展方法

（2）理解 P0 口上地址和数据信号分离的手段

2. 实验器材：

（1）G2010+实验平台	1	台	（2）G6W 仿真器	1	台
（3）连线	若干	根	（4）编程器	1	台
（5）27256 芯片	1	片	（6）万用表	1	块
（7）紫外线擦除器	1	台			

3. 实验原理：

MCS-51 单片机应用系统中，当程序长度大于片内 ROM 空间，或单片机（如 8031）内部没有 ROM 时，就需要扩展外部程序存储器，这时单片机的/EA 引脚必须接地。单片机在访问外部程序存储器时由 P2 口送出地址的高 8 位，P2 口有输出锁存功能，可直接与外部存储器芯片的地线相接，而 P0 口则分时用作地址、数据总线，分别用作输出低 8 位地址和输入指令，这时单片机通过地址锁存允许信号 ALE 和/PSEN 信号指示 P0 口的信息类别，换言之，当 ALE 信号当由高电平变成低电平时刻，P0 口上的信息为低 8 位地址 A0-A7，当/PSEN 信号由低电平变成高电平时刻，P0 口上的信息为输入的指令代码，因此当扩展程序存储器时，需要先用锁存器将出现在 P0 口的低 8 位地址进行锁存，然后将锁存后的地址与外部存储器芯片的地址线相接。锁存器一般都是由一组带三态门控制的 D 触发器组成，常见的如 Intel8282,74LS373，74LS573 等，它们都有一个输入选通端：74LS373 的 G 端，8282 的 STB 端，当选通信号为高电平时，输出端 Qi 信号随输入端 Di 的变化而变化，当选通端的信号由高电平变成低电平时，输出端 Qi 的信号即被锁定，与输入端 Di 的信号无关，通常将 ALE 信号与锁存器的选通端相连，通过锁存器将 P0 口上的地址和数据信息分离。常用的程序存储器芯片有两种,即 EPROM 和 EEPROM 型。EPROM 型主要指由紫外线擦除的 27 系列芯片如:2716、2732、2764、27128、27256、27512 等。EEPROM 型主要指由电擦除的 28 系列芯片，如 5V 电压写入的 2816A、2817A 以及 2864 等。程序存储器芯片与单片机连接时，数据线接 P0 口，地址线与地址总线相连，/OE 接单片机的/PSEN，/CE 接译码器或接地，具体线路如下图所示。

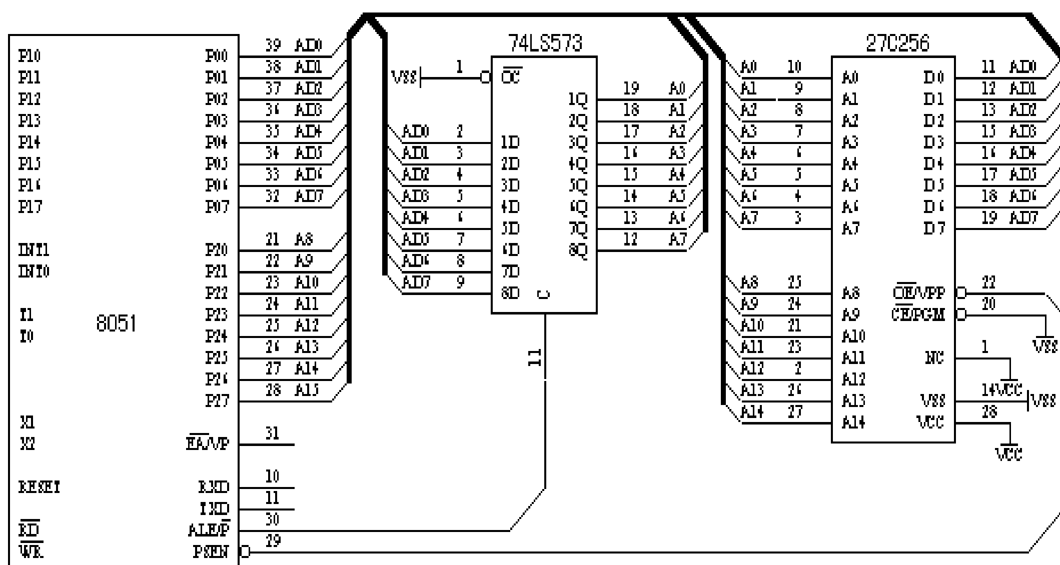
4. 实验步骤：

（1）使用 G2010+实验平台上锁紧插座，按图示搭建好硬件电路，把 G2010+实验平台上的监控芯片 27C256 拔下，插入锁紧插座。按上图连接好硬件。

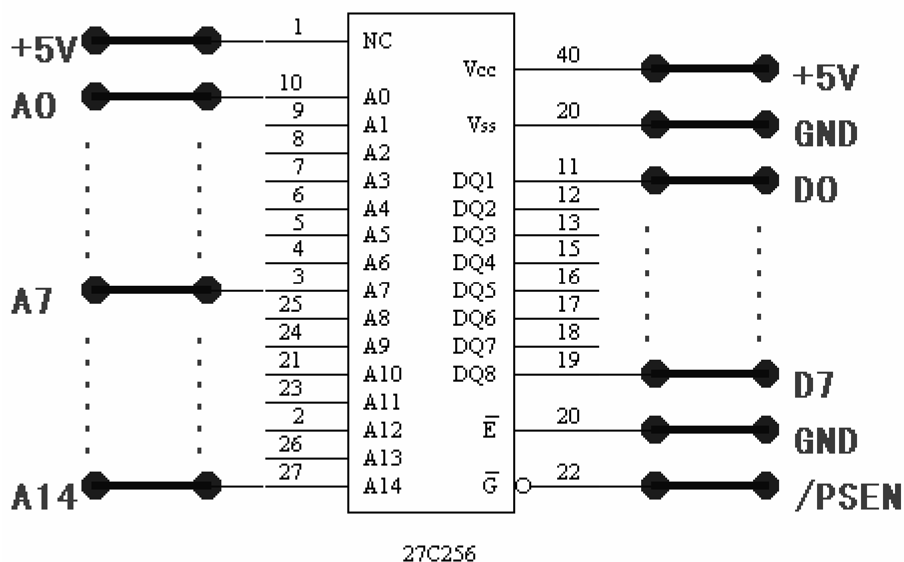
（2）用仿真器来测试硬件有否故障：把 G6W 仿真器与 G2010+实验平台连接好，设置仿真模式为 4，即程序存储器和数据存储器均指向用户空间。初始化后，观察 CPU 窗口的代码是否与原监控一致。若读到的值是 00，01，02，03，04……，则表明：EPROM 的/CE、/OE 脚的信号不对；若读到的值部分有误，则可根据规律查核地址线、数据线是否有误。

（3）无故障，则卸下仿真器，上电，观察结果是否与监控一致。

5. 接线方案：



EPROM27C256 扩展实验原理图



EPROM27C256 扩展实验接线图

6. 程序清单:(MCS51\C02.ASM)

实验三　静态数据存储器扩展实验

1．实验目的：

掌握 MCS-51 单片机系统扩展数据存储器的方法；掌握 SRAM6264 与 8031 的接口方法

2．实验内容：

编写程序在 6264 的 1000H-1000AH 单元中分别写入 01H、02H、……0AH，这 10 个数作为一延时 500ms 子程序的入口参数以实现 0.5 秒到 5 秒钟的不同时间延时，结果通过 P1.2 上连接的二极管的亮、灭状态显示出来。

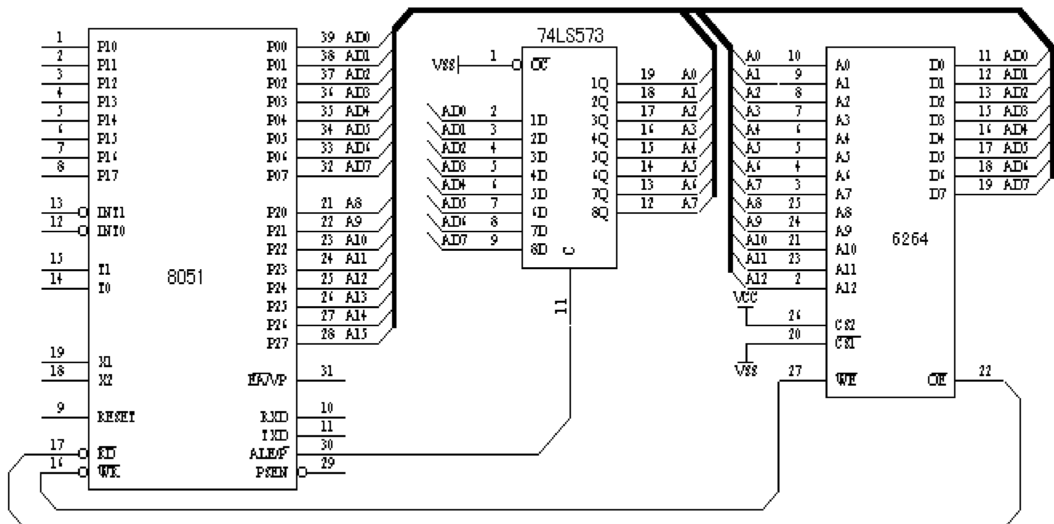
3. 实验器材：

（1）G2010+实验平台	1	台	（2）G6W 仿真器	1	台	（3）连线	若干	根
（4）6264 芯片	1	片	（5）万用表	1	块			

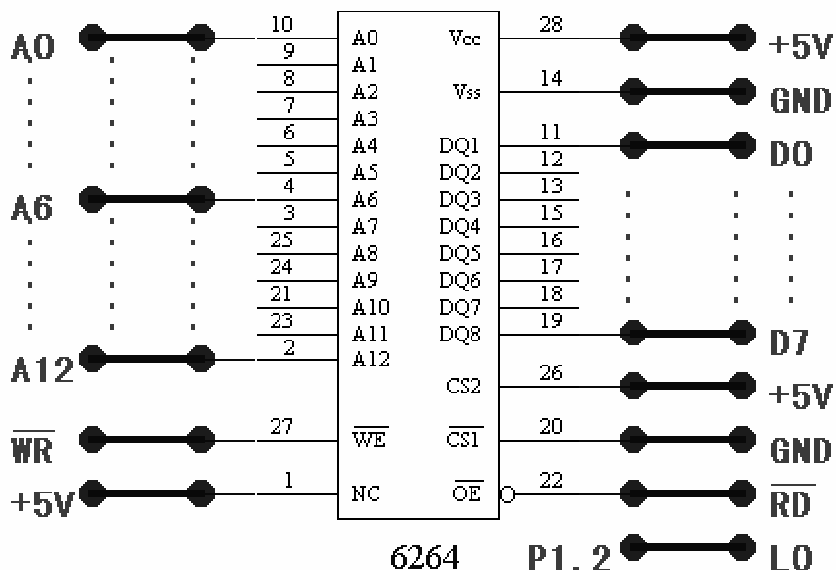
4．实验原理：

8031 单片机内部具有 128 字节的 RAM,它们可以作为寄存器、堆栈以及 RAM 使用，是十分珍贵的资源，但在许多实际的应用系统中，仅芯片内部的 RAM 往往不够，这时就必须外扩数据存储器，最多不超过 64KB。由于动态存储器外围电路复杂，在单片机系统中扩展的 RAM 一般为静态 RAM、EEPROM 或 FLASH MEMORY，常用的静态 RAM 有 6116(2KX8)、6264(8KX8)、62256（32KX8）等，数据存储器芯片的引脚，主要包括地址线（A0-An）、数据线（D0-D7）、片选（/CE）、输出允许（/OE）和写（/WE）控制信号线，数据存储器的扩展就是地址、数据和控制三组总线与 MCU 的连接，通常 A8-A10 直接由 P2 口提供，A7-A0 来自地址锁存器；数据线（D0-D7）接 P0 口；输出允许（/OE）引脚接系统的读（/RD）信号 P3.7,写信号（/WE）接系统的写信号（/WR）P3.6，片选（/CE）由系统的译码器产生，它决定了数据存储器芯片在系统的地址。

5.接线图案：

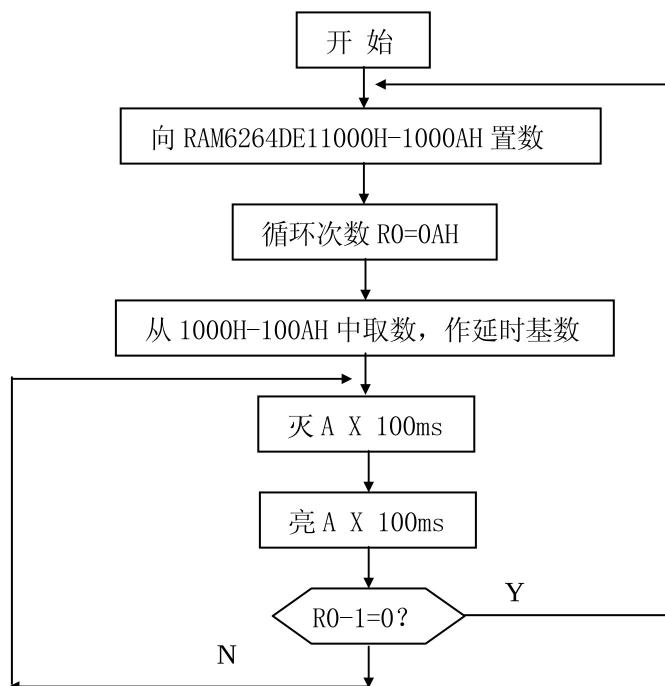


RAM6264 扩展实验原理图



RAM6264 扩展实验接线图

6. 程序框图：



7. 实验步骤：

（1）使用 G2010+实验平台上锁紧插座，把 6264 芯片插入锁紧插座，按图搭建好硬件电路。

（2）把 G6W 仿真器与 G2010+实验平台连接好，设置仿真模式为 2，即程序存储器留在仿真器，数据存储器指向用户空间。

（3）硬件诊断：

（3．0） G2010+G6W 连 PC 机，在 WINDOWS 调试环境下点击外部数据窗口图标（ ），在打开的 XDATA 窗口中的 0000H 开始的地址上（即 RAM6264 的空间）上写入一串数据，然后点击刷新图标（ ），若显示的数据无变化则表明硬件无故障。如显示的数据变为 00H，则表明/WR、/CE、/RD、VCC、VSS 未接好；如显示的数据变为有规律的其它数据，则数据线有问题；如显示的数据，在某些地址上相同，则可能地址线有问题。

（3．1） G2010+G6W 连 PC 机，在 DOS 调试环境下，把光标移入 XDATA 窗中，按 Alt+G，在打开的“Goto Address”窗口中输入“0000H”，再按“回车”键，在 0000H 地址上打入数据，如显示的数据与打入的数据一致， 则表明硬件无故障。

（3．2）如选购了 G2K 仿真板，则还可直接在 G2010 实验平台的键盘上进行硬件诊断：设定工作模式为模式 2，即数码管显示为“PI EE”，然后按“MON”键使数码管显示为“ ’ ”，输入“0000”，按“+”，再输入“55”，如数码管显示的数据也是“55”，则表明硬件无故障。

（4）编写程序并调试，观察结果。

8．程序清单：（MCS51\C03.ASM）

实验四　并行 I/O 口扩展实验

1．实验目的：

（1）掌握 MCS-51 单片机系统扩展 I/O 口的方法。

2. 实验内容：

利用 74LS244 作为输入口，读取开关状态，并将此状态，通过 74LS273 再驱动发光二极管显示出来。

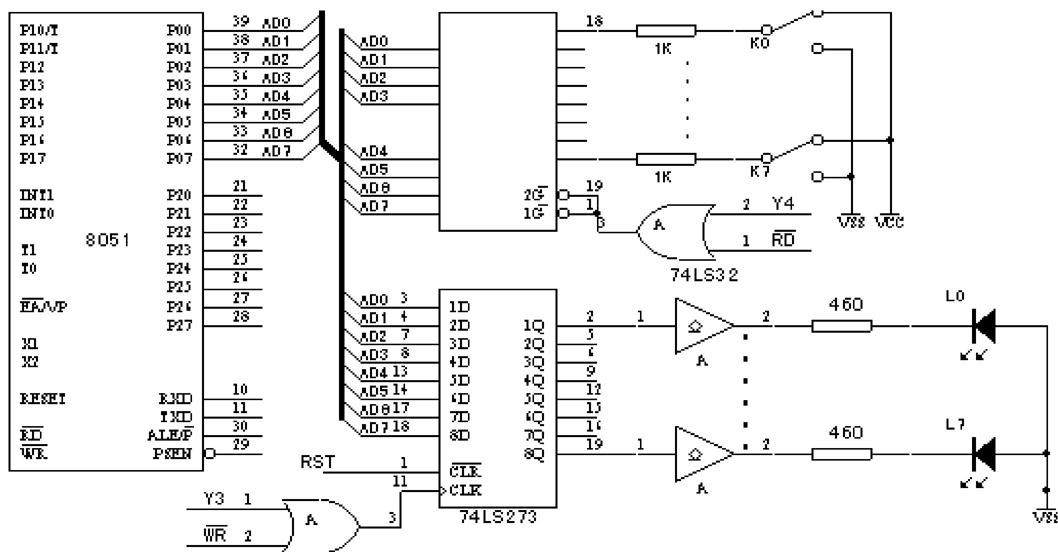
3．实验器材：

（1）G2010+实验平台	1	台	（2）G6W 仿真器	1	台
（3）连线	若干	根	（4）74LS244 芯片	1	片
（5）74LS273 芯片	1	片	（6）万用表	1	块

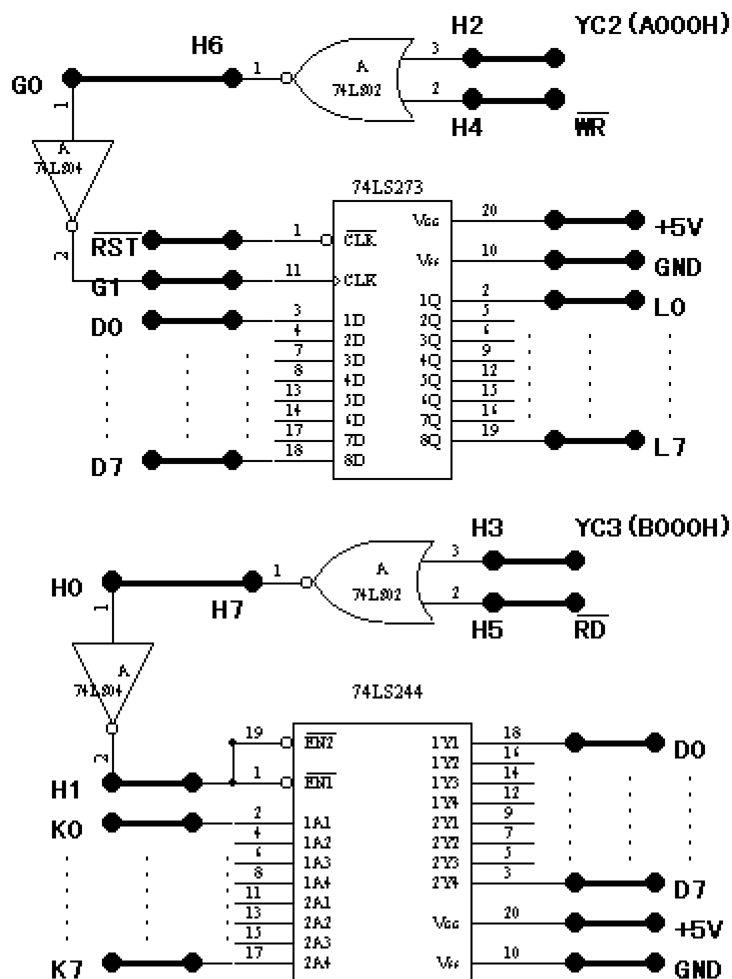
4．实验原理：

8031 单片机从理论上讲具有 4 个 I/O 口即 P0、P1、P2 和 P3，而实际使用时，P0 分时复用为地址总线的低 8 位和数据总线，P2 用作地址总线的高 8 位，P3 经常用作第二功能，真正作为并行 IO 口使用的只有 P1 口。在许多实际的应用系统中，常需要扩展 I/O 口。I/O 接口电路一般具有输出锁存，输入缓冲，与总线隔离等功能，单片机系统中扩展并行 I/O 口一般有三种方法，（1）采用专用的并行 IO 接口芯片如 8155，8255 等实现；（2）通过串行口外接移位寄存器芯片如 74LS164、74LS165 等实现；（3）通过外接锁存器芯片 74LS377、74LS273 等芯片，并经过总线隔离驱动芯片 74LS245、74LS244 隔离实现。本实验中采用的正是上述的第三种方法。

5. 接线图案：

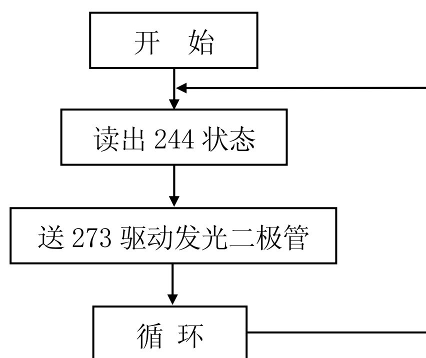


并行 IO 口扩展实验原理图



并行 IO 口扩展实验接线图

6. 程序框图：



7. 实验步骤：

（1）设定工作模式为模式 2，即程序存储器在仿真器上，数据存储器指向用户板。使用了实验平台锁紧插座。按接线方案连接电路。

（2）硬件测试：

（2．0） G2010+G6W 连 PC 机，在 WINDOWS 调试环境下点击外部数据窗口图标（ ），

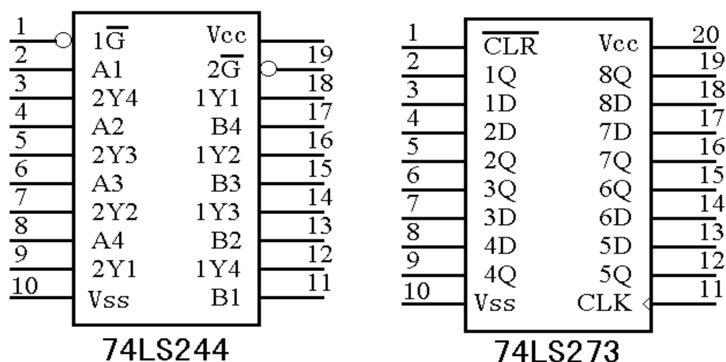
在打开的 XDATA 窗口中的 0A000H 地址（即 74LS273 的片选空间）上写入 55H 或 0AAH，则输出发光二极管也应显示 55H 或 AAH，否则，74LS273 这一块硬件电路有问题。然后点击刷新图标（ ），观察 0B000H（即 74LS244 的片选空间）地址上的值，应与拨动开关状态一致，否则，74LS244 这一块硬件电路有问题。

（2．1） G2010+G6W 连 PC 机，在 DOS 调试环境下，把光标移入 XDATA 窗中，按 Alt+G，在打开的“Goto Address”窗口中输入“0B000H”，再按“回车”键，在 0A000H 地址上打入 55H 或 0AAH，则输出发光二极管也应显示 55H 或 AAH，否则，74LS273 这一块硬件电路有问题。观察 0B000H（即 74LS244 的片选空间）地址上的值，应与拨动开关状态一致，否则，74LS244 这一块硬件电路有问题。

（2．2）如选购了 G2K 仿真板，则还可直接在 G2010+实验平台的键盘上进行硬件诊断：设定工作模式为模式 2，即数码管显示为“PI EE”，然后按“MON”键使数码管显示为“ ’ ”，输入“A000”，按“+”，再输入 55H 或 0AAH，则输出发光二极管也应显示 55H 或 AAH，否则，74LS273 这一块硬件电路有问题。按“MON”键，数码管显示返回“P”，输入“B000”，按“+”，读出的值应与拨动开关状态一致，否则，74LS244 这一块硬件电路有问题。

（3）编写并执行程序，拨动 K0-K7，观察 L0-L7 是否对应点亮。

8. 器件引脚:



9. 思考问题：

（1）可否用 273 输入，244 输出。

（2）当输入开关为奇数个 1 时，发光二极管状态循环点亮，当输入开关为偶数 1 时，发光二极管正确对应输入开关状态，程序怎样修改。

10. 软件清单： (MCS51\C04.ASM)

实验五　串行口扩展并口实验

1. 实验目的：

（1）掌握 MCS-51 单片机串行口方式 0 时的工作原理。

（2）了解方式 0 时的应用，即通过串行口扩展输出口，进行静态显示的方法

（3）掌握串行移位寄存器芯片 74LS164 的工作原理。

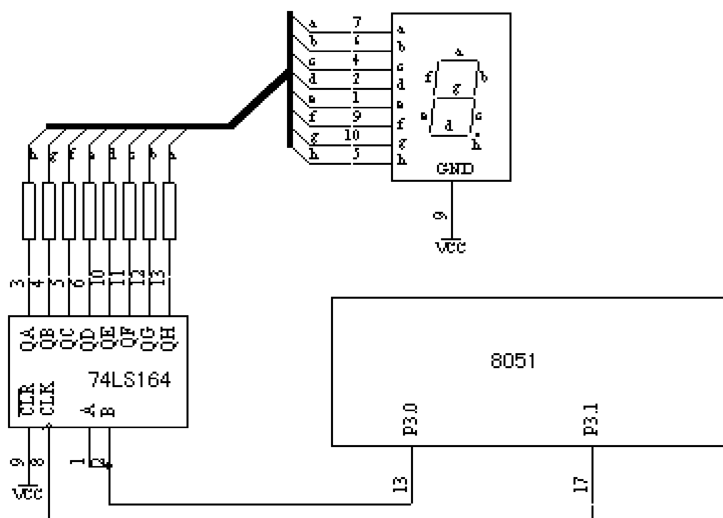
2. 实验内容：

编制程序使数码管循环依次显示 0-f。

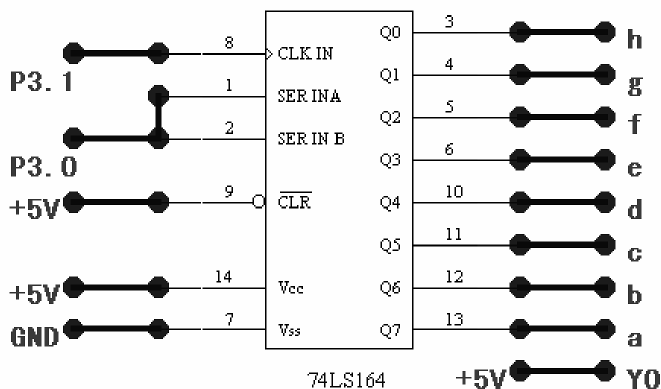
3. 实验器材：

（1）G2010+实验平台	1	台	（2）G6W 仿真器	1	台
（3）连线	若干	根	（4）74LS164 芯片	1	片

4. 接线图案：

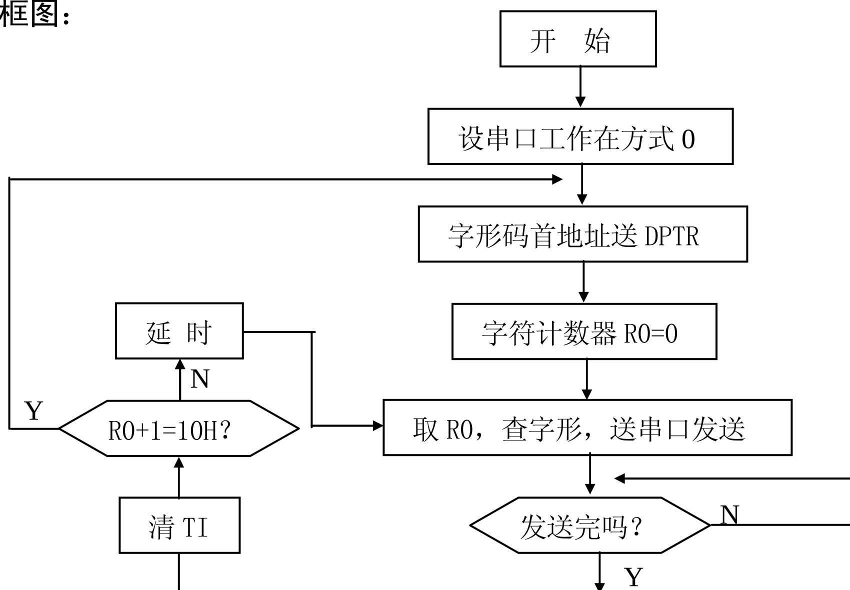


串行口扩展实验原理图



串行口扩展实验接线图

5. 程序框图：



6. 实验原理：

MCS-51 单片机除了具有四个 8 位并行口外，还具有一个全双工的串行通信接口，该接口有 4 种工作方式，当工作于方式 1、2 和 3 时，作 UART（通用异步接收和发送器)，用以实现单片机系统之间点对点的单片机通信、多机通信和单片机系统与 PC 机之间的通信；而工作于方式 0 时，为同步移位寄存器输入/输出方式，常用于扩展 IO 口，这时串行数据通过 RxD 引脚输入或输出，引脚 TxD 输出同步移位信号，收/发数据都为 8 位，低位在前，波特率为振荡频率的 1/12。本实验中通过 74LS164 实现串入并出。

7. 实验步骤：

(1)把 G2010+实验平台上 U10(74LS240)上边的跳线座用跳线器短接起来,使 74LS240 工作起来。拔下 74LS164(U9,数码管右侧),插入 G2010+锁紧插座,,按图示搭建好硬件电路。

（2）设置仿真模式为 2，即程序存储器留在仿真器，数据存储器指向用户空间。

8. 软件清单： (MCS51\C05.ASM）

实验六　多个外中断源扩展实验

1. 实验目的：

掌握 MCS-51 单片机系统有多个外中断源时的处理方法。

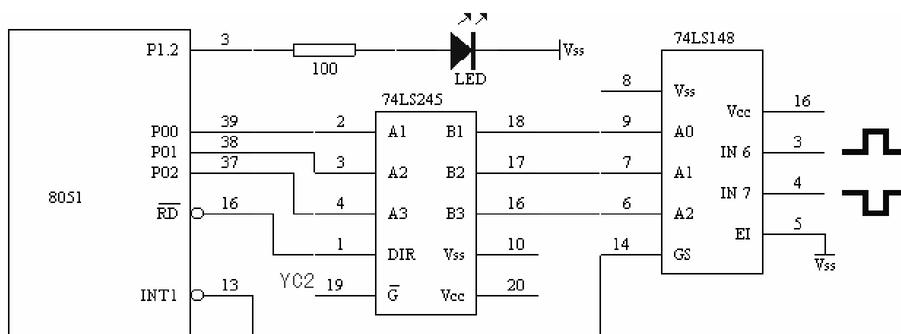
2. 实验器材：

（1）G2010+　1 台　（2）G6W 仿真器　1 台　（3）连线　若干　根　（4）74LS148　1　片

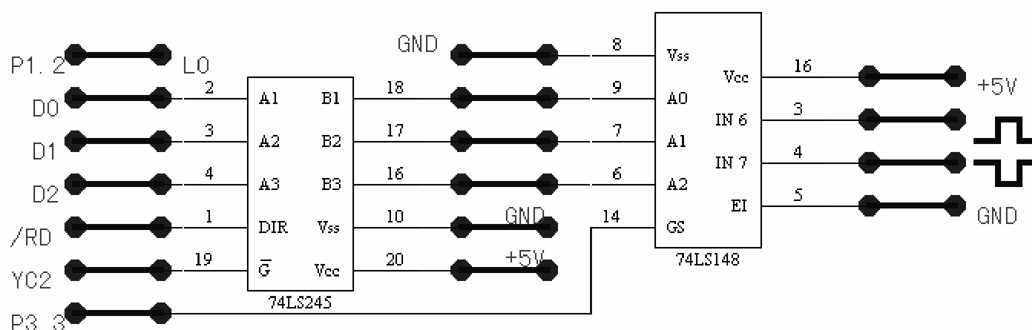
3. 实验原理：

扩展外部中断源，一般有三种办法：（1）编程 T0 和 T1 工作在对外计数方式，并设计数初值为计数的最大值，这样只要引脚出现一个脉冲，就可以使 T0/T1 溢出产生中断；（2）将多个中断源通过一个“与门”连接到/INT1 上，它们同时也被连接至某一输入口如 P1，这样每当/INT1 产生中断时，通过查询输入口即可知道是哪个中断源提出了中断请求，当响应速度要求很高时，这种方法不适应；（3）通过优先权编码器 74LS148 来扩展外部中断源。74LS148 有 8 个外部中断源输入端 IRQ0-ITQ7，IRQ7 的优先权最高，3 个编码端 A2-A0，一个编码群输出端 GS，一个使能端/EI。当/EI 为低电平时，只要其个输入端中任意有一个输入为低电平，GS 端输出就为低，且在 A2-A0 输出一组相应的编码，当有多个输入端同时输入为低时，A2-A0 输出优先权最高的中断源对应的编码，当 IRQ7-IRQ0 分别有中断请示时输出的编码分别为 000、001、010、011、100、101、110 和 111。

4. 接线方案：

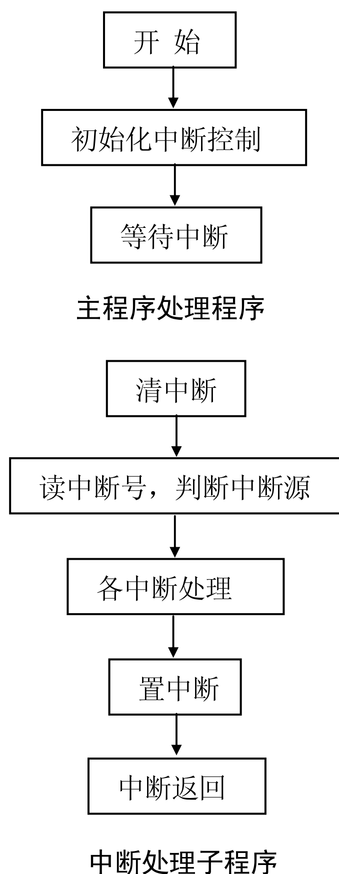


多个外中断源扩展实验原理图



多个外中断源扩展实验接线图

5．程序框图：



中断处理子程序

6．实验步骤：

（1）使用 G2010+实验平台上的锁紧插座，分别插上 74LS245 和 74LS148，按图示搭建好硬件电路。

（2）把 G6W 仿真器与 G2010+实验平台连接好，设置仿真模式为 2，即程序存储器留在仿真器，数据存储器指向用户空间。

（3）编写主程序，初始化为低电平触发中断，且发光管灭；编写两个中断服务程序，当 TR2 按下时，发光管烁 5 次，间隔 250ms，退出中断程序时，使发光管灭；当 TR2 松开时，发光管闪烁 10 次，间隔 250ms，退出中断程序时，使发光管灭。

7．程序清单：(MCS51\C06.ASM)

实验七　MCS-51 单片机与 IBM-PC 机通信实验

1. 实验目的：

（1）掌握 MCS-51 单片机串行口工作于 UART 方式时的工作原理和编程方法。

（2）掌握 PC 串行通信程序的编程方法。

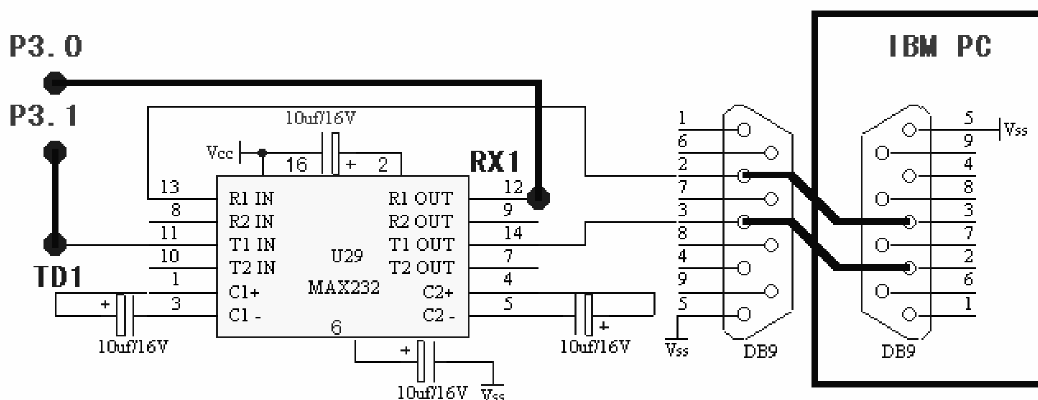
2. 实验器材：

（1）G2010+实验平台	1　台	（2）G6W 仿真器	1　台
（3）连线	若干　根	（4）计算机	1　台

3. 实验原理：

MCS-51 单片机的串行口，当工作于方式 1、2 和 3 时，作 UART（通用异步接收和发送器）使用，以实现单片机系统之间点对点的单机通信、多机通信和单片机系统与 PC 机之间的通信；PC 机串行通信主要是通过串行接口芯片 8251 的实现的。8251 有 10 个寄存器，端口地址从 3F8H-3FE（COM1），可以通过对 8251 的编程来指定通信协议即通信的波特率、数据位数、奇偶类型和停止位长度，具体办法请参考微机原理教材。另外 PC 机串口的电平为 RS-232 电平，而 MCS-51 串口的电平是 TTL 电平，要想实现两者之间的通信，需要在它们之间加电平转换电路，传统的方法是使用 1488 将 TTL 电平转换成 RS-232 电平，用 1489 实现反向转换，由于 1488 需要±12V 电压，使用中很不方便，故我们的实验中用 MAX232 代替，MAX232 为单一+5V 电源供电，既可实现 TTL 到 RS-232 的电平转换，也可实现 RS-232 到 TTL 电平的转换，使用十分方便，具体线路如下图所示。

4. 接线图案：

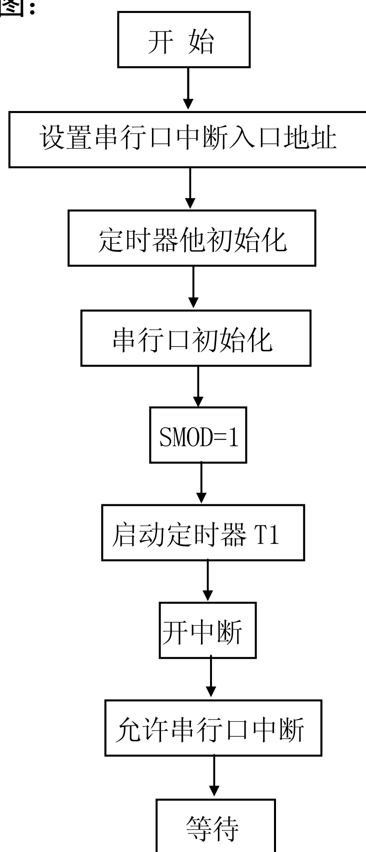


5. 实验步骤：

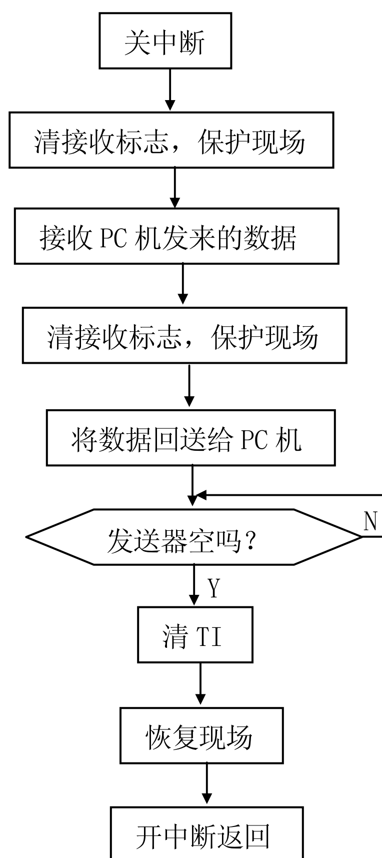
（1）编制 PC 机通信程序。通信协议为波特率 1200，1 个起始位，8 个数据位，1 个停止位。程序运行时，通过 PC 机键盘输入一个字符，然后将字符的 ASCII 码发送给单片机，再从单片机接收该字符，收到后在 PC 机显示器上显示该字符。

（2）编制单片机通信程序。通信协议为波特率 1200，1 个起始位，8 个数据位，1 个停止位。程序运行时，通过串行口接收由 PC 机发送的字符，收到后将接收到的字符的 ASCII 码通过发光管显示出来，然后将该字符再发送给 PC 机。

6．程序框图：



主程序框图



中断子程序框图

7．实验提示：

（1）本实验使用的计算机需有两个串行口，一个供调试用，另一个供与单片机通信用。

（2）本机提供的 PC 机例程用汇编程序编写。应先利用 MASM 编译软件进行编译，然后用 LINK 生成 EXE 文件。本文件在 DOS 下执行。

（3）本实验中上位机软件使用的是串口 1，故应把仿真器接在串口 2 上。

8．程序清单：（MCS51\C07.ASM）（上位机:MCS51\PC.ASM）

实验八　8155 接口芯片使用实验

1．实验目的：

（1）掌握 MCS-51 单片机系统 IO 口扩展方法。

（2）掌握并行接口芯片 8155 的性能以及编程使用方法。

（3）了解软件、硬件调试技术。

2.实验内容：

编写并调试出一个程序，其功能为对 8155 初始化，使 8155 PA 口为输入口，PB 口为输出口，并把一组数据写入 8155 内部 RAM，当输入开关为全 0 时，则按顺序把 8155 内部 RAM 数据读出显示，当输入开关为非全 0 时，则直接把开关状态在发光二极管上显示出来。

3．实验器材：

（1）G2010+实验平台　　1　　台

（2）G6W 仿真器　　1　　台

（3）连线　　若干　　根

（4）8155 芯片　　1　　片

4．实验原理：

MCS-51 单片机的 4 个 I/O 口中，通常情况下，只有 P1 口作为 I/O 口使用，实际使用中经常需要扩展 I/O 口，扩展 I/O 口方法之一就是采用专用的 I/O 接口芯片如 8155，8255 等。本实验中采用 8155 扩展了两个输出口、一个输入口以实现键盘输入和数码管输出。可编程并行接口芯片 Intel 8155 内部含有 256 字节的静态 PAM，两个并行 8 位口 PA、PB，一个并行的 6 位口 PC，以及一个 14 位的定时/计数器，是单片机系统最常用的接口芯片之一，掌握其性能与作用方法非常重要。

8155 可直接与 CPU 接口,8155 的 RAM 和 I/O 编址由 IO/M 和 ALE 锁存的地址来控制,IO/M=0 选择 RAM，编址为 00-FFH；IO/M=1 对/8155 的 I/O 口进行读写，8155 内部 I/O 编址如下：

A_2	A_1	A_0	I/O 口
0	0	0	命令状态口
0	0	1	PA 口
0	1	0	PB 口
0	1	1	PC 口
1	0	0	定时器低 8 位
1	0	1	定时器高 6 位和方式

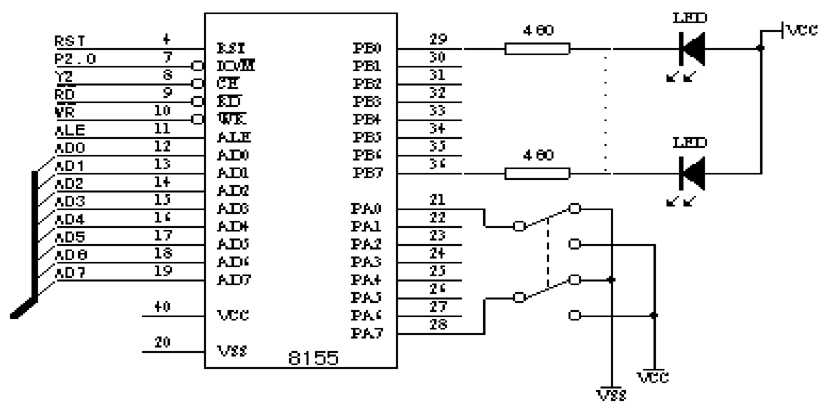
8155 的命令字如下：

D7	D6	D5	D4	D3	D2	D1	D0
TM_2	TM_1	IEB	IEA	PC_2	PC_1	PB	PA

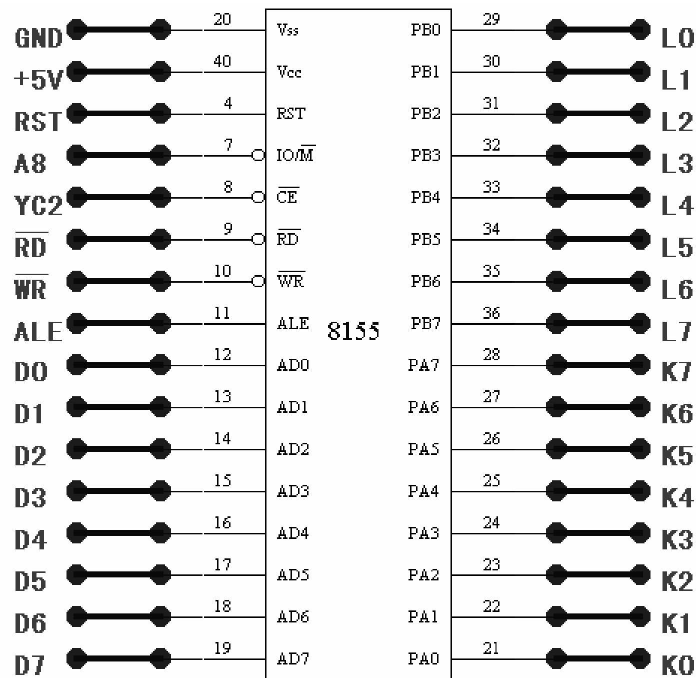
- PA、PB：定义口 A、口 B，0 为输入，1 为输出

- IEA、IEB：口 A、口 B 中断控制，1 为允许，0 为禁止
- PC1、PC2：定义口的工作方式
- TM1、TM2：定时器命令具体实验线路如下图所示，实验时需将实验板上的 P1 和 P2、P4 和 P6 连接起来。

5．接线图案：



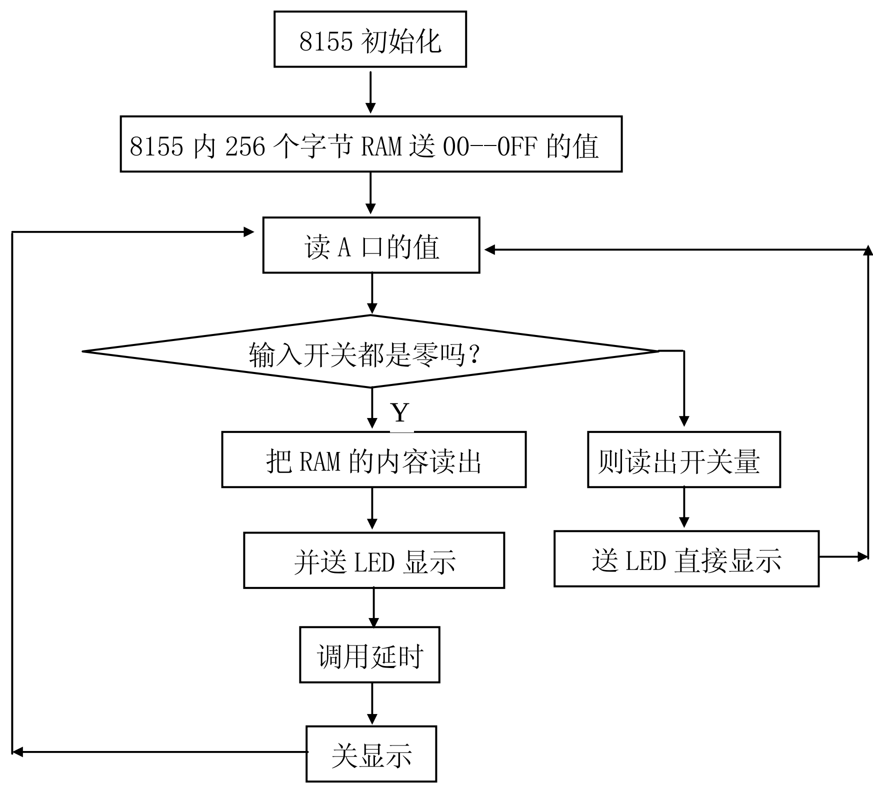
8155 接口芯片使用实验原理图



8155 接口芯片使用实验接线图

本实验中 8155 编程为 A 口输入，B 口、C 口输出，根据状态控制字选择方法得到 8155 方式字：02H。

6. 程序框图：



程 序 流 程 图

7. 实验步骤：

（1）设定工作模式 2，即程序存储器在仿真器上，数据存储器指向用户板。8155 的命令口为 0A100H、A 口为 0A101H、B 口为 0A102H、C 口为 0A103H、定时器低 8 位定时器为 0A104H、高 6 位定时器为 0A105H、8155 内部 RAM 的地址为：0A000H-0A0FFH。

（2）硬件测试：

（2．0） G2010+G6W 连 PC 机，在 WINDOWS 调试环境下点击外部数据窗口图标（ ），在打开的 XDATA 窗口中的 0A100H 地址（即 8155 的片选空间）上写入控制字 02H，在 0A102H（8155 的 B 口）上打入 55H，0AAH 等数字，观察 LED 的状态是否正确。在 0A100H 以后的地址上输入 55H、0AAH 等数字，然后点击刷新图标（ ）,刚输入的数据如有变化，则表明硬件有问题。

（2．1） G2010+G6W 连 PC 机，在 DOS 调试环境下，把光标移入 XDATA 窗中，按 Alt+G，在打开的“Goto Address”窗口中输入“0A100H”，再按“回车”键，在 0A000H 地址上打入“02H”，对 8155 进行初始化，在 0A102H（8155 的 B 口）上打入 55H，0AAH 等数字，观察 LED 的状态是否正确。

（2．2）如选购了 G2K 仿真板，则还可直接在 G2010+实验平台的键盘上进行硬件诊断：设定工作模式为模式 2，即数码管显示为“PI EE”，然后按“MON”键使数码管显

示为“ ’ ”，输入“A100”，按“+”，再输入“02”，对8155进行初始化，再在

“A102”（8155的B口）上打入55H，0AAH等数字，观察LED的状态是否正确。

（3）编写程序、编译程序，用单步、断点、连续方式调试程序，排除软件错误。运行程序，观察输入开关和输出指示灯状态，直至达到本实验的要求为止。

8.思考问题：

（1）试编写程序，当输入开关状态为奇数个1时，输出口则对应输入口状态，当输入为偶数个1时为全0。

（2）试编写程序，使用8155定时器，每隔1秒依次读出RAM数据，在PB口发光二极管上显示。

9.软件清单：（MCS51\C08.ASM）

实验九　键盘、显示接口芯片 8279 使用实验

1．实验目的：

掌握 8279 的工作原理以及编程方法；掌握单片机系统中扩展 LED 和键盘的接口技术。

2．实验内容：

利用定时器 1，设计时钟程序。

3．实验器材：

（1）G2010+实验平台　1　台
（2）G6W 仿真器　1　台
（3）连线　若干　根
（4）8279 芯片　1　片

4．实验原理：

单片机系统中使用 LED 有两种方法，即静态显示和动态显示，静态显示的优点是显示效果好，编程简单，但由于输出的每位都需要锁存，使用的硬件较多；静态显示方式中，各位数码管的 a-h 端并连在一起，每一时刻只有一位数码管被点亮，各位依次轮流被点亮，硬件电路简单，但由于需要经常进行刷新显示和扫描键盘输入，降低了 CPU 的效率，而且编程的工作量很大。为了解决动态显示中存在的问题，Intel 公司研制出了专用的键盘显示器接口电路芯片 8279，该芯片可以实现对键盘的自动扫描，识别闭合键的键号，完成显示器的动态显示，应用非常方便。8279 用 A0 来区分信息特征，当 A0 为 0 时，CPU 从 8279 读出是状态，写入的是命令，且每个命令也有自己的特征；当 A0=1 时读出和写入的都是数据。8279 内部有两个缓冲区，即一个 8 字节的先进先出 FIFO 键盘 RAM 和一个 16 字节的显示 RAM，显示数据时只要将待显示数据的段码写入显示 RAM 即可；当有键闭合时，8279 会自动执行去抖、得到键值、等待按键释放等操作，最后将键值存入 FIFO RAM 中，程序只需从 FIFO 中读取键值即可，编程十分简单。

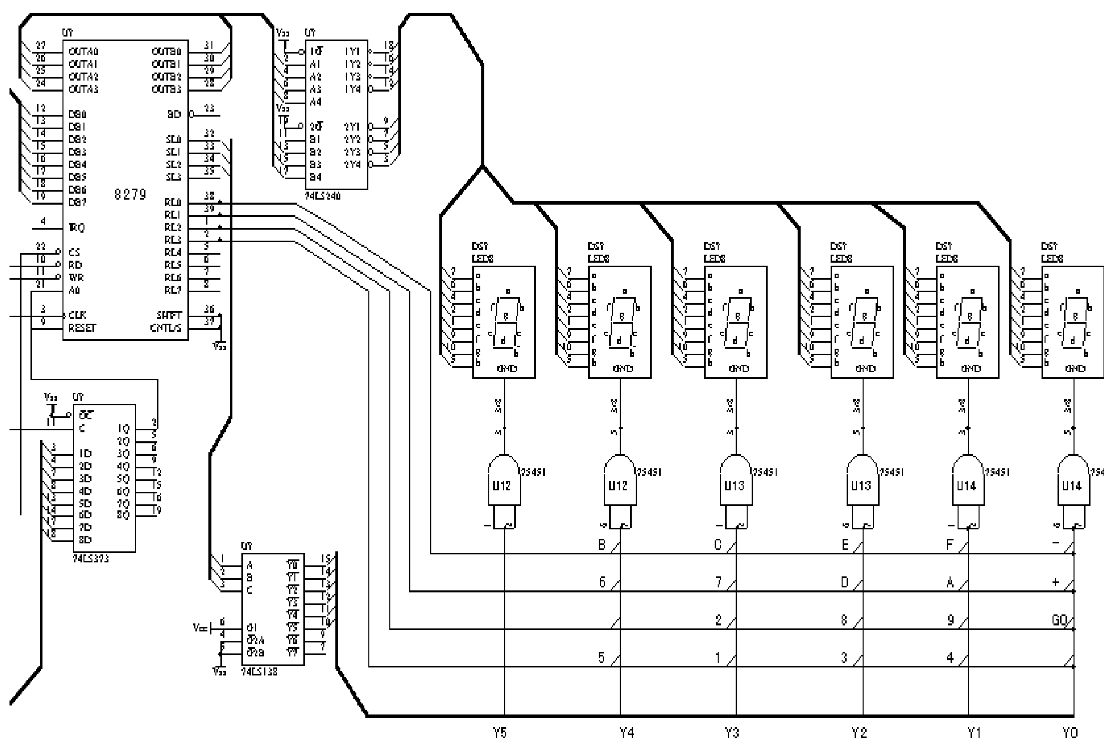
5．接线图案：

①8279 是一通用可编程键盘显示接口器件，与 LED 数码管、驱动器、按键可组成一个最基本键盘显示系统。在这里 8279 外接 4 位数码管作为输出设备。SL0-SL3 外接 Y3-Y0 孔，对 LED 显示器进行译码方式扫描，8279 CUTA-CUTB 作为显示器的段数据输出口，连接a－h孔。

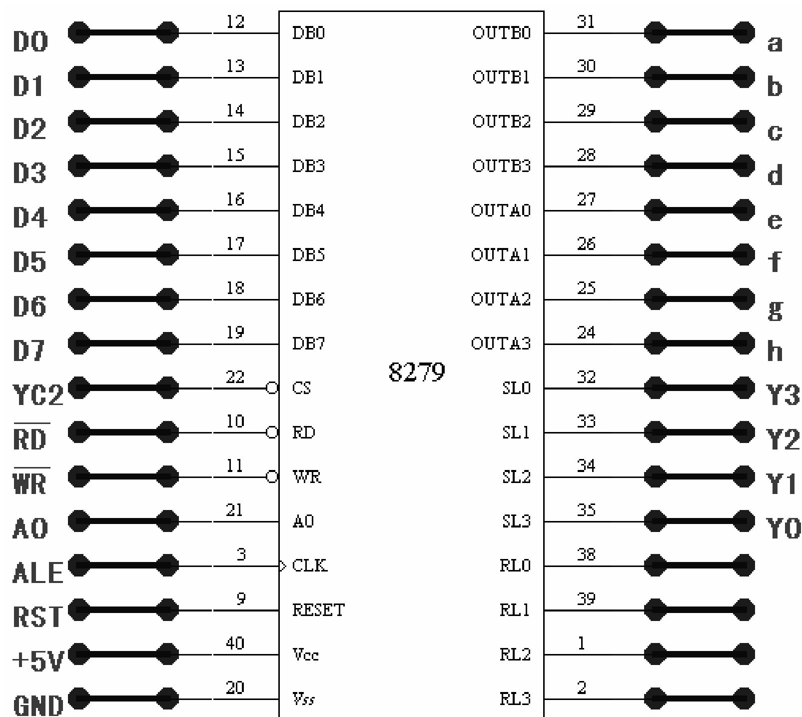
②从原理图可以看出 138 译码器的 Y2（0A000H-0AFFFH）作为 8279 选片端，低位地址 A0 作为 8279 A0 选择线。所以 8279 命令/状态口地址：0A000H，数据口地址：0A001H。

6．程序框图：

请参阅《模放式实验例程：实验十四　电子时钟（定时/计数器定时实验）中的程序框图，Page70



键盘、显示接口芯片 8279 使用实验的原理图



键盘、显示接口芯片 8279 使用实验的接线图

7．实验步骤：

（1）使用G2010+实验平台上实验锁紧插座,插上8279。把G2010+实验平台上U10(74LS240)上边跳线座用跳线器短接起来,使74LS240工作起来。拔下74LS164(U9,数码管右侧)。

（2）设置仿真模式为2，即程序存储器留在仿真器，数据存储器指向用户空间。

（3）硬件测试：

（3．0） G2010+G6W连PC机，在WINDOWS调试环境下点击外部数据窗口图标（ ），在打开的外部数据（XDATA）窗口中的0A000H地址（即8279的命令/状态口地址）上分别写入0EH、32H、0DCH （对8279进行初始化），在0A001H地址上写入0C0H（“0”字的字形），数码管应显示“0”。

（3．1） G2010+G6W连PC机，在DOS调试环境下，把光标移入XDATA窗中，按Alt+G，在打开的“Goto Address”窗口中输入“0A000H”，再按“回车”键，在0A000H地址上分别打入“0EH、32H、0DCH ”（对8279进行初始化），在0A001H地址上写入0C0H（“0”字的字形），数码管应显示“0”。

（3．2）如选购了G2K仿真板，则还可直接在G2010实验平台的键盘上进行硬件诊断：设定工作模式为模式2，即数码管显示为“PI EE”，然后按“MON”键使数码管显示为“ ’ ”，输入“A000”，按“+”，再分别输入0EH、32H、0DCH，对8279进行初始化，按“+”键，在“0A001H”地址上输入“0C0H”，数码管应显示“0”

（3）用单步、断点、连续断点、连续方式调试程序，检查程序运行结果是否正确，排除软件错误，观察按键功能与数码管显示状态，直至达到本实验的要求为止。

8．思考问题：

修改程序，使得按“8”键显示“8”字循环、按“9” 键显示“9”字循环、按“A”键显示“A”字循环。

9．程序清单：（C09.ASM）

实验十　8255 控制交通灯实验

1．实验目的：

（1）掌握 8255 的工作原理以及编程方法，了解软件与硬件的调试技术。

2．实验内容：

编写并调试出一个实验程序，其功能为用 8255 模拟控制交通灯运行情况。

3．实验器材：

（1）G2010+实验平台　1　台　　（2）G6W 仿真器　1　台

（3）连线　若干　根　　（4）8255 芯片　1　片

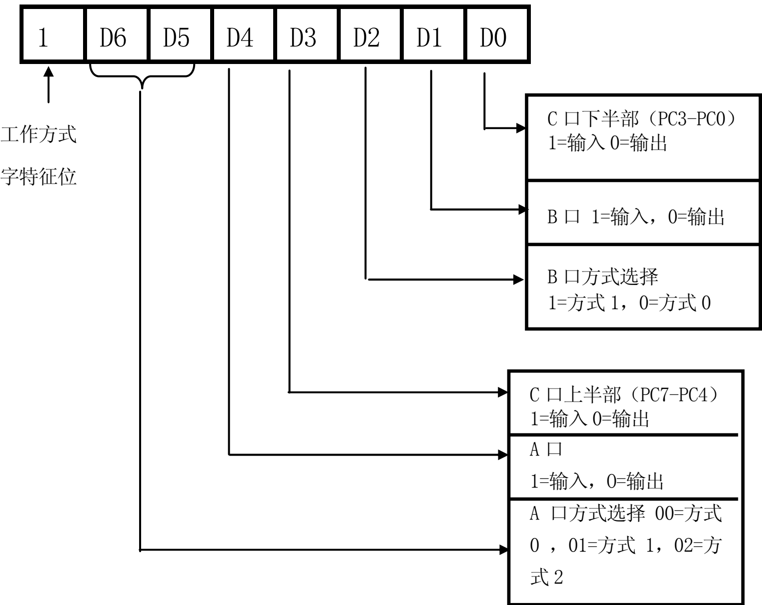
4．实验原理：

8255 有三个 8 位的并行口，端口既可以编程为普通 I/O 口，也可以编程为选通 I/O 口和双向传输口。8255 为总线兼容型，可以与 8051 的总线直接接口。其中，口地址取决于片选 CS 和 A1、A0。

选择如下：

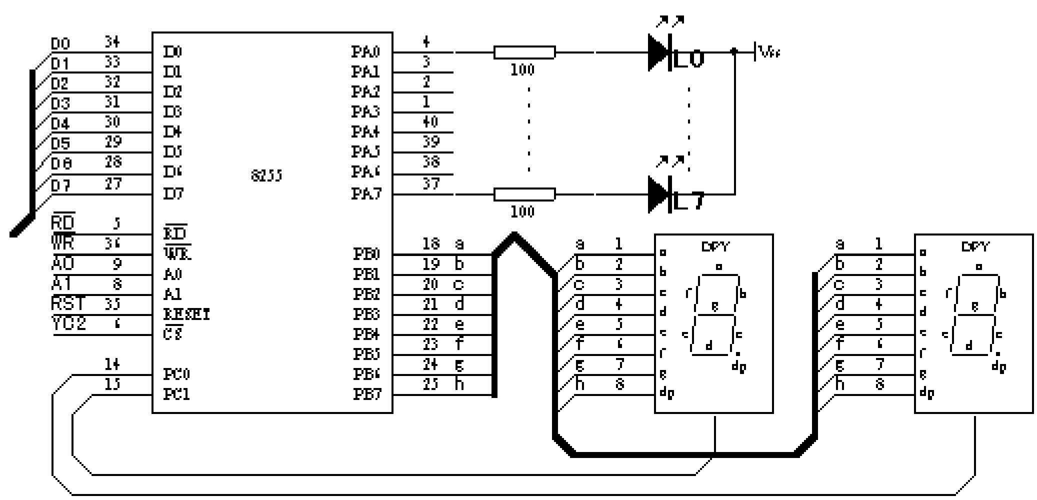
A1	A0	CS	选通的端口
0	0	0	A 口
0	1	0	B 口
1	0	0	C 口
1	1	0	命令口

8255 方式字选择：

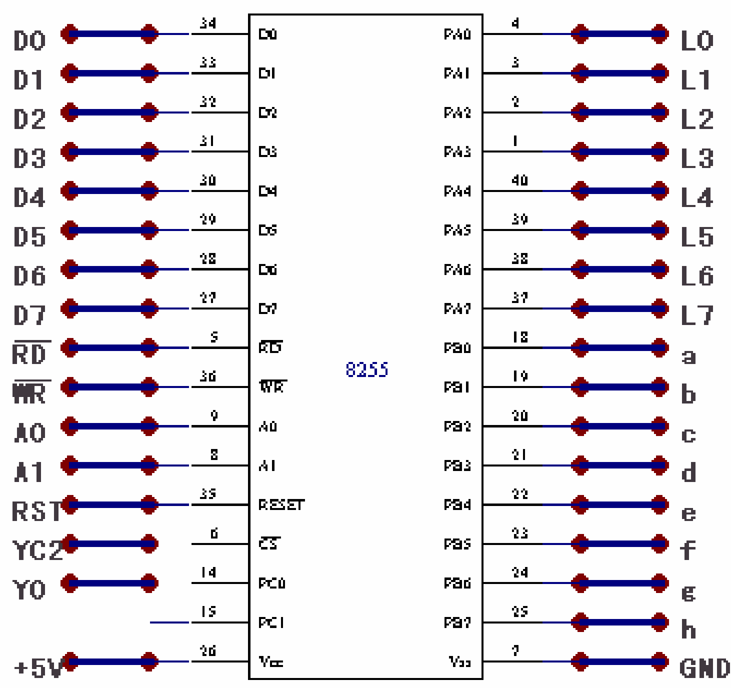


本实验中 8255 编程为 PB 口、PC 口、PA 口均输出，根据 8255 状态控制字选择方法。8255 控制字应为 80H。

5. 接线方案：



8255 控制交通灯实验原理图



8255 控制交通灯实验接线图

6. 实验步骤：

（1）设定仿真器工作模式为模式 2，即程序存储器在仿真器上，数据存储器指向用户板。使用实验仪锁紧插座，如图连线，因 8255 片选信号为 0A000H，所以，A 口为 0A000H、B 口为 0A001H、C 口为 0A002H、命令口为 0A003H。

（2）硬件测试：

（2．0） G2010+G6W 连 PC 机，在 WINDOWS 调试环境下点击外部数据窗口图标（ ），

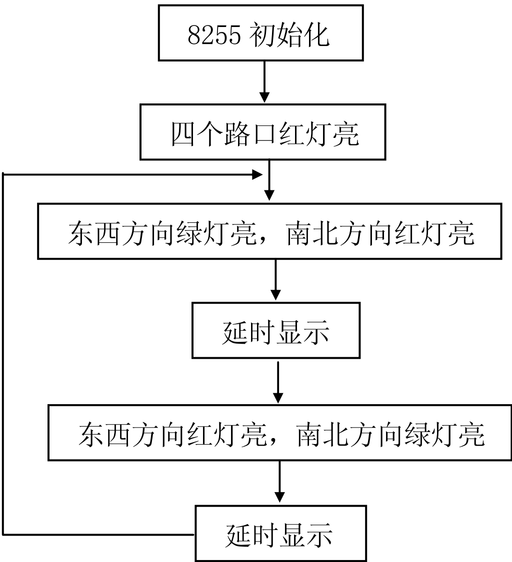
在打开的 XDATA 窗口中的 0A003H 地址（即 8255 的片选空间）上写入控制字 80H，在 0A000H（8255 的 A 口）上打入 55H，0AAH 等数字，用逻辑笔观察 8255 的 PA 口状态是否为 55H、AAH。若不是，则硬件有问题。

（2．1） G2010+G6W 连 PC 机，在 DOS 调试环境下，把光标移入 XDATA 窗中，按 Alt+G，在打开的“Goto Address”窗口中输入“0A003H”，再按“回车”键，在 0A003H 地址上打入“80H”，在 0A000H（8255 的 A 口）上打入 55H，0AAH 等数字，用逻辑笔观察 8255 的 PA 口状态是否为 55H、AAH。若不是，则硬件有问题。

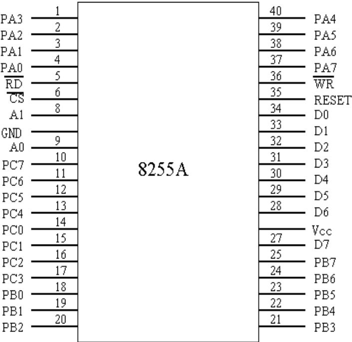
（2．2）如选购了 G2K 仿真板，则还可直接在 G2010 实验平台的键盘上进行硬件诊断：设定工作模式为模式 2，即数码管显示为“PI EE”，然后按“MON”键使数码管显示为“ ’ ”，输入“A003”，按“+”，再输入“80”，对 8255 进行初始化，再在“A000”（8255 的 C 口）上打入 55H，0AAH 等数字，用逻辑笔观察 8255 的 PA 口状态是否为 55H、AAH。若不是，则硬件有问题。

（3）编写程序、编译程序，用单步、断点、连续方式调试程序，排除软件错误。运行程序，直至达到本实验的要求为止。

7. 程序框图



8、器件引脚：



9、软件清单：（MCS51\C10.ASM）

实验十一　可编程计数/定时器 8253 实验

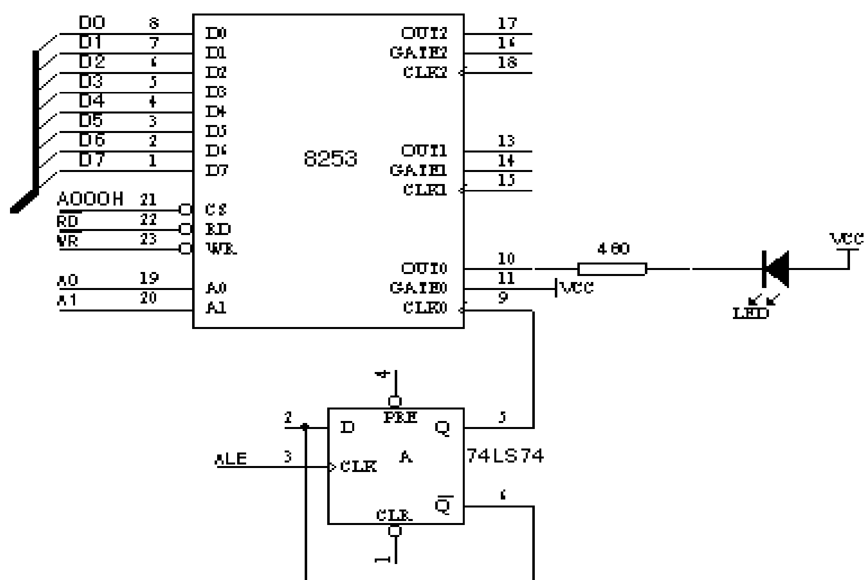
1．实验目的：

掌握单片机与 8253 硬件接口设计方法，了解 8253 定时器/计数器工作原理，熟悉对 8253 初始化编程。

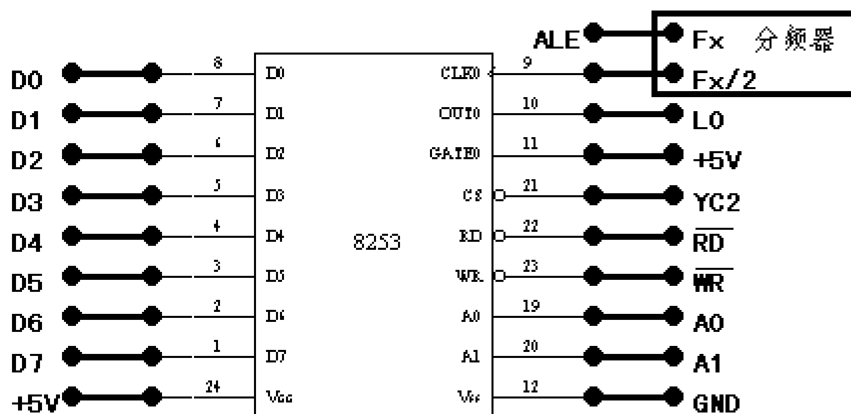
2．实验内容：

编写并调试出一个实验程序，其功能对 8253 初始化，使得 T0 工作于方式 3，产生方波，驱动发光二极管。

3．实验线路：



可编程计数/定时器 8253 实验的原理图



可编程计数/定时器 8253 实验的接线图

4．实验原理：

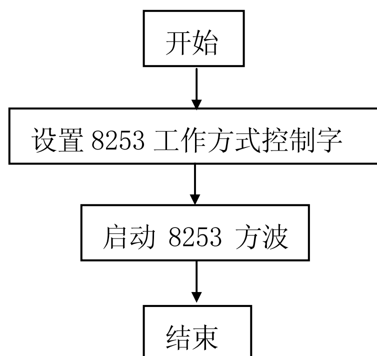
（1）8253 有 3 个计数器，CLK0 用 ALE 进行分频输出频率信号作为输入。

（2）8253 的输出 OUT0 经驱动后接发光二极管。

（3）从实验原理图可以看出 74LS138 译码器 YC2 作为 8253 选片端，A0、A1 作为 8253 端口地址选择线，所以 8253 端口地址为：计数器 0：0A000H　　方式控制口：0A003H

（4）确定 8253 T0 方式字：计数器 T0 先读写低位字节，后读写高位字节，工作于方式 3，二进制计数，T0 方式字为 36H。

5．程序框图：



6．实验步骤：

（1）设定仿真工作模式为模式 2，即程序存储器在仿真器上，数据存储器指向用户板。

（2）按实验线路图连线。

（3）硬件诊断：

（3．0） G2010+G6W 连 PC 机，在 WINDOWS 调试环境下点击外部数据窗口图标（ ），在打开的外部数据（XDATA）窗口中的 0A003H 地址（即 8253 的命令口地址）上写入方式字 36H，对 8253 进行初始化，在 0A000H 地址（计数器 T0）上先写入 00H（低位字节），后写入 0FFH（高位字节），观察指示灯应以一定频率变化，否则，硬件有故障。

（3．1） G2010+G6W 连 PC 机，在 DOS 调试环境下，把光标移入 XDATA 窗中，按 Alt+G，在打开的“Goto　Address”窗口中输入“0A003H”，再按“回车”键，在 0A003H 地址上分别打入“36H”（对 8253 进行初始化），在 0A000H 地址（计数器 T0）上先写入 00H（低位字节），后写入 0FFH（高位字节），观察指示灯应以一定频率变化，否则，硬件有故障。

（3．2）如选购了 G2K 仿真板，则还可直接在 G2010 实验平台的键盘上进行硬件诊断：设定工作模式为模式 2，即数码管显示为“PI EE”，然后按“MON”键使数码管显示为“ ’ ”，输入“A003”，按“+”，再输入 36H，对 8253 进行初始化，按“-”键，在“A000”地址上输入“00”，再按“-”键，再按“+”键，还是在“A000”地址上输入“FF”，再按“-”键，指示灯应以一定频率变化。

（4）输入程序，汇编成目标程序，用单步、断点、连续方式调试程序，检查程序运行结果是否正确，可用示波器检查 8253 计数器输出端的变化波形。

7．程序清单：（C11.ASM）

实验十二　串行 EEPROM 93C46 扩展实验

1．实验目的：

掌握串行 EEPROM 93C46 与单片机的接口方法和编程方法；了解三线串行通信技术。

2．实验器材：

（1）G2010+实验平台　1　台　　（2）G6W 仿真器　1　台

（3）连线　　若干　根　　（4）93C46 芯片　1　片

3．实验原理：

串行 EEPROM 因其具有体积小、功耗低、价格便宜，且掉电后数据不丢失等优点近年来得到了广泛的应用。本实验将使同学们掌握 ATMEL 公司生产的 EEPROM93C46 的使用方法。93C46 的引脚分布如图所示。CS 为片选信号，高电平有效，在连续的二条指令之间，CS 必须保持至少 250μS 的低电平；SK 为时钟信号，所有操作码、地址码、数据位均在 SK 信号的上升沿输入或输出，SK 信号的最高频率为 1MHz；DI 为串行数据输入端；DO 为串行数据输出端，并在擦/写周期提供忙/闲状态；ORG 为低电平时，输出 8 位数据，为高电平时，输出 16 位数据；GND 接地；Vcc 接 5V 电源。对 93C46 的操作共有 7 条指令如下表所示。

指令	起始位	操作码	地址 8	地址 16	数据 8	数据 16	说明
READ	1	10	A6-A0	A5-A0			从指定单元读出数据
EWEN	1	00	11XXXXX	11XXXX			擦写指定单元
ERASE	1	11	A6-A0	A5-A0			擦除指定单元
WRITE	1	01	A6-A0	A5-A0	D7-D0	D15-D0	写入指定单元
ERAL	1	00	10XXXXX	10XXXX			片擦除
WRAL	1	00	01XXXXX	01XXXX	D7-D0	D15-D0	片写入
EWDS	1	00	00XXXXX	00XXXX			擦/写禁止

读 READ 指令的功能是从指定单元中读出数据，其时序如图 1（a）所示。擦写允许指令 EWEN 的时序如图 1（b），在执行擦/写指令前执行该指令。

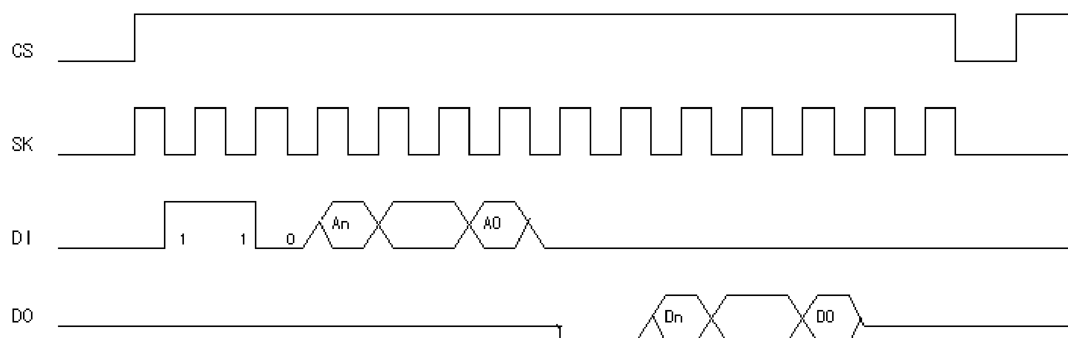


图 1（a）读时序图

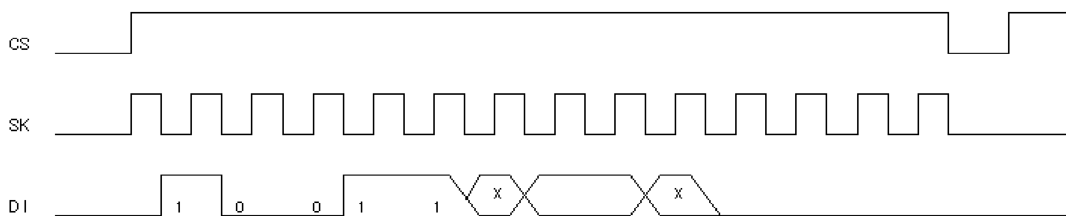


图 1（b）擦/写允许时序图

擦除指令 ERASE 完成对指定单元擦除，即使对应单元置 1，其时序如图 1（c）所示。写指令完成对指定单元实现写数据，其时序图如图 1（d）所示。对于擦除指令和写指令，当 DO 为高电平时表明操作结束。

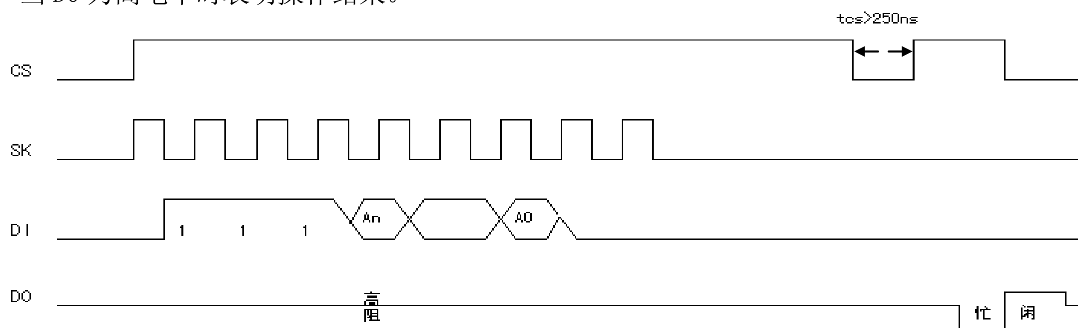


图 1（c）擦除时序图

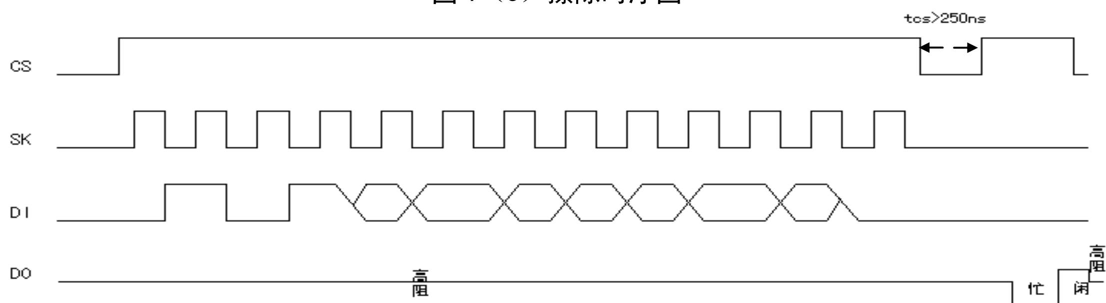


图 1（d）写时序图

片擦指令 ERAL 使所有位置 1，片写指令 WRAL 实现对整个芯片写入指定的数据，擦/写禁止指令 EWDS 禁止对芯片进行擦/写，它们的操作时序图如图 1 的 e、f 和 g。

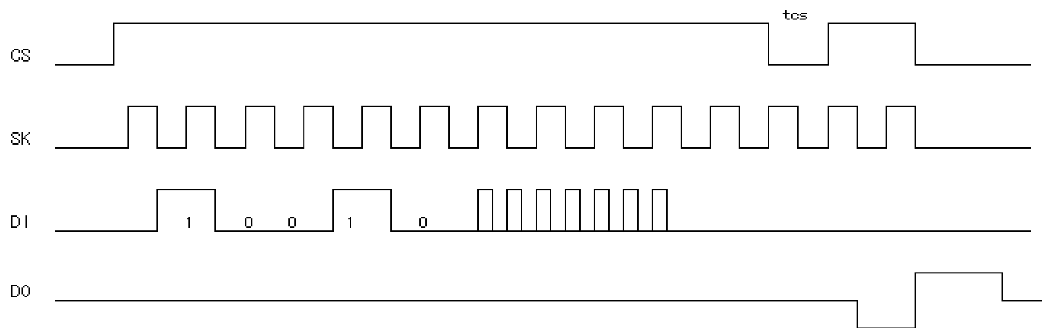


图 1（e）　片擦时序图

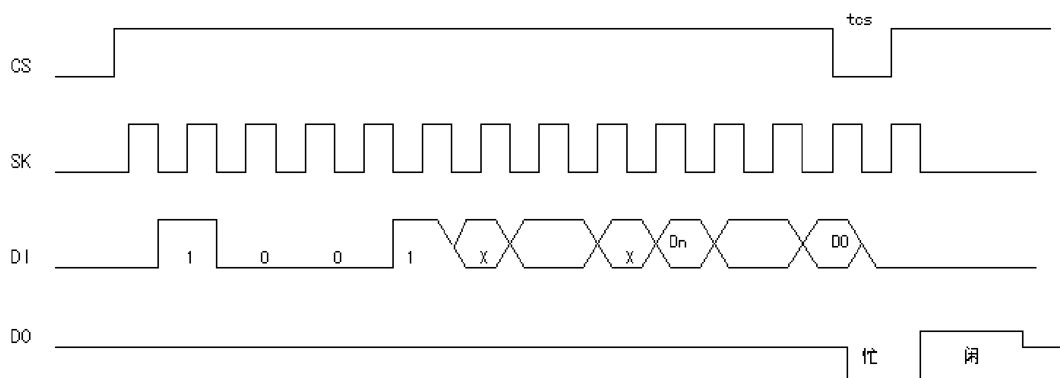


图 1（f）　片写时序图

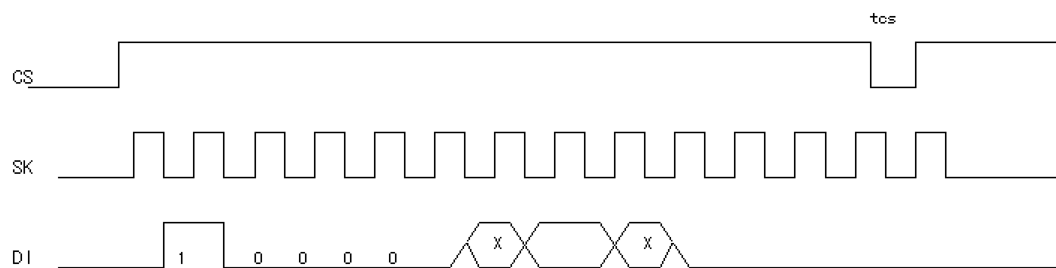
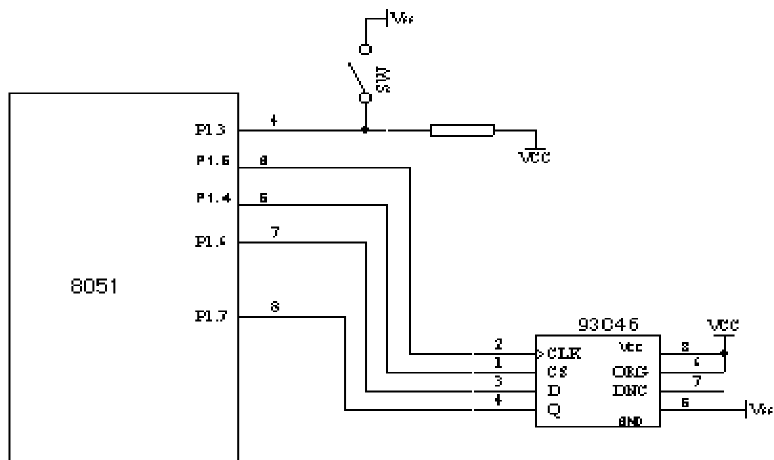
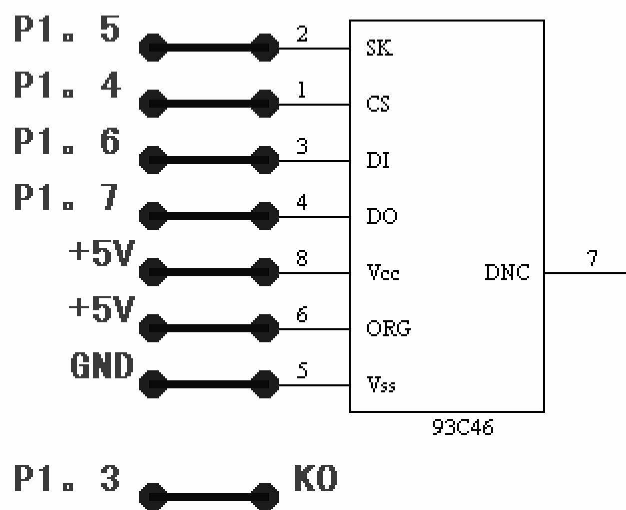


图 1(g)　擦写禁止时序图

4．接线图案：



93C46 的引脚分布以及与 8031 的接口原理图



93C46 与 8031 的接口接线图

5.实验步骤：

编写程序，当图中的 K0 与 VCC 相连时将字符串“412186”写入 93C46 中；当 K0 与地相连时从 93C46 中读出字符串并通过数码管显示出来。

6.软件清单：(MCS51\C12.ASM)

实验十三　串行 EEPROM AT24C01 扩展实验

1．实验目的：

（1）　掌握串行 EEPROM AT24C01 与 51 单片机的接口方法和编程方法。

（2）　了解I^2C总线接口技术。

2．实验器材：

（1）G2010 实验平台　1　台　　（2）G6W 仿真器　1　台

（3）连线　　　　若干　根　　（4）AT24C01 芯片　1　片

3．实验原理：

I^2C（Inter IC BUS）总线是PHILIPS公司推出的设备内部串行总线，它由一根数据线SDA和一根时钟线SCL组成，SDA和SCL都为双向I/O 线，通过上拉电阻Ｒp接+5V电源，总线空闲时皆为高电平，I^2C总线的输出端必须是开漏或集电极开路，以便具有“线与”功能，总线数据传输速率为 100Kbps。具有I^2C总线的设备都是工作在主从方式，由主设备发开始（START）和停止（STOP）信号，串行时钟线SCL的上升沿将数据写入从设备，下降沿将数据从从设备读出，即在SCL为高电平时，SDL为高电平期间SDA的下降沿为开始信号，而SDA的上升沿为停止信号。ATMEL公司的串行EEPROM AT24CXX（AT24C01、02、04、08、16）就是属于这种接口的芯片，它们的容量分别是 128*8、256*8、512*8、1K*8、2K*8。对AT24C系列EEPROM的操作有字节写、页写、现行地址读、随机读和序列读等方式，实现字节写的时序为：

S	1	0	1	0	A2	A1	A0	0	A	addr	A	data	A	P

S 起始信号，由主节点提供；SDA 在 SCL 高电平期间由高电平变成低电平；1010 为器件型号地址；A2A1A0 页地址；0 为写操作；A 应答信号，在主从节点之间任何字节传送完后，从节点必须有应答信号，肯定时应答信号为低电平，否定时应答信号为高电平；addr 地址字节，指定片内操作单元地址。Data 为数据字节，由主节点出发，从节点接收；P 停止信号，SDA 在 SCL 为高电平期间由低电平变成高电平。如将数据#55H 写入 20H 单元的时序为：

S	10100000	A	00100000	A	01010101	A	P

随机读是按指定地址读出一个单元的数据，其时序为：

S	1	0	1	0	A2	A1	A0	0	A	addr	A	S	1	0	1	0	A2	A1	A0	1	A	data	/A	P

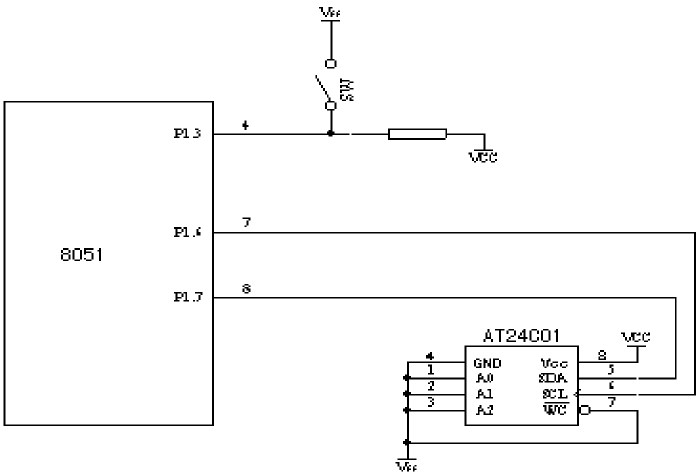
如将 30H 单元中的数（#10H）读出的时序为：

S	10100000	A	30H	A	S	10100001	A	10H	/A	P

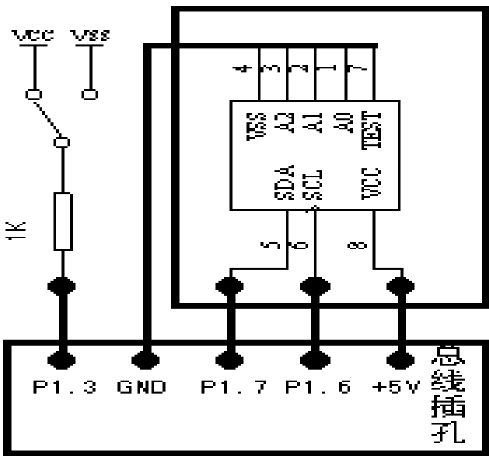
对AT24CXX的其它操作请参考有关文献，这里不再叙述。实验中使用的串先EEPROM为AT24C01，因MCS-51 单片机没有I^2C接口，故用P1.6 和P1.7 两条IO 线来模拟实现I^2C总线，具体

实验线路如图所示。

4．接线图案：



串行 EEPROM AT24C01 扩展实验原理图



串行 EEPROM AT24C01 扩展实验接线图

5.实验步骤：

编写程序，当图中的 K0 打开时将字符串“412186”写入 AT24C01A 从 10H 开始的单元中；当 K0 闭合时从 93C46C 中读出字符串并通过数码管显示出来。

6．软件清单：(MCS51\C13.ASM)

实验十四　AT89C2051 控制步进电机

1、实验目的：

学习使用 AT89C2051 进行简单控制；学习程序目标代码固化和脱机运行方法。

2、实验内容：

利用 89C2051 定时器 0 定时，以控制步进电机正反运行。

3、实验器材：

（1）G2010+实验平台　 1　台

（2）G6W 仿真器　　　 1　台

（3）连线　　　　　　若干　根

（4）AT89C2051 芯片　1　 片

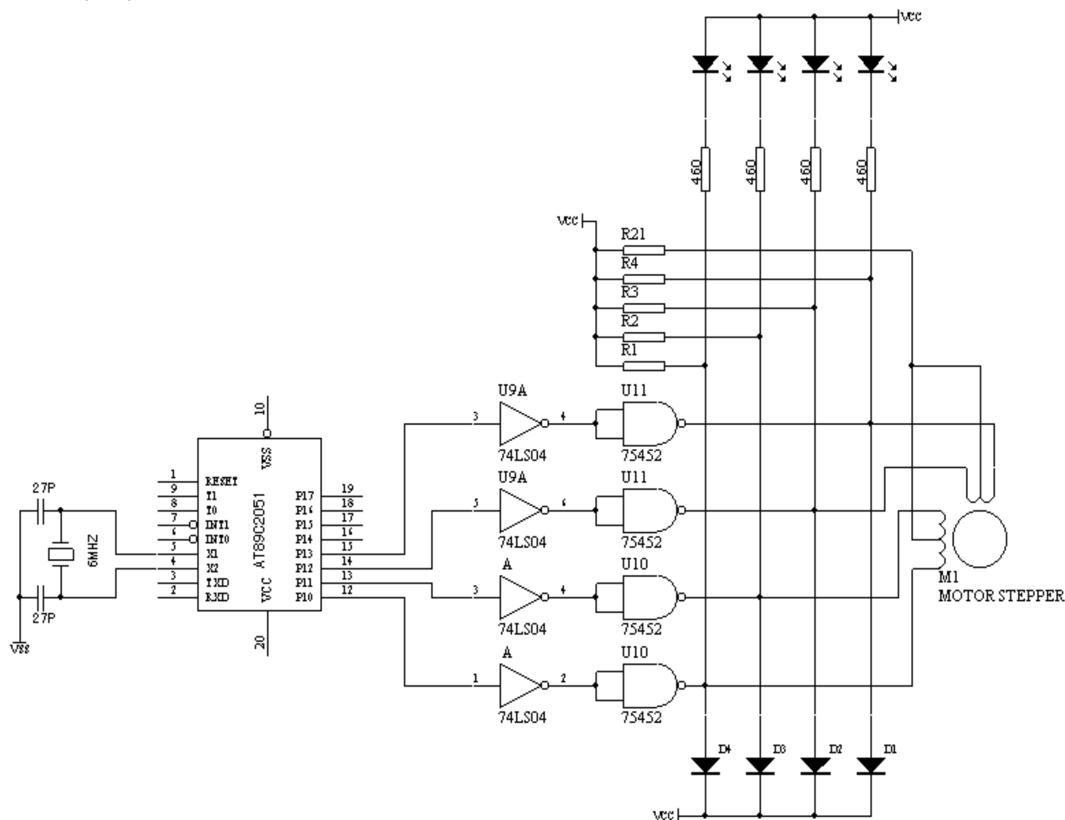
（5）通用编程器　　　 1　台

（6）使用步进电机电路

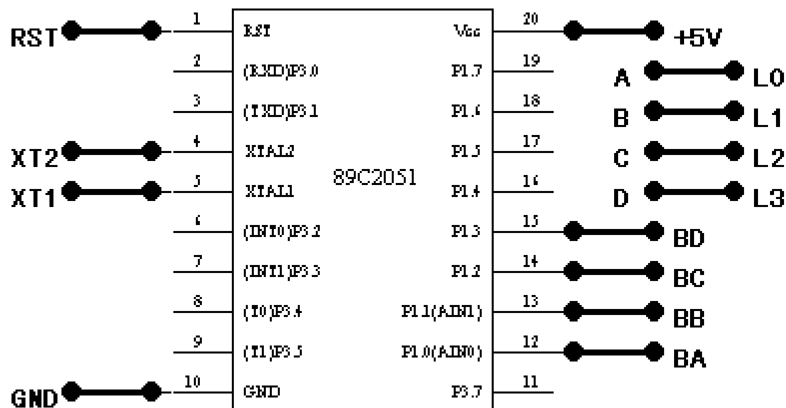
（7）使用锁紧插座及晶振源、RST 电路等辅助电路

（8）需另配 POD2051 转换适配器

4、接线图案：



AT89C2051 控制步进电机实验原理图



AT89C2051 控制步进电机实验接线图

5、实验步骤：

（1）把 LAB8051CPU 板取下

（2）把POD2051 配在POD51 仿真头上，然后插入锁进插座中，按实验接线图连线，连步进电机电路，设计程序，使电机转动，L_0-L_3闪亮。

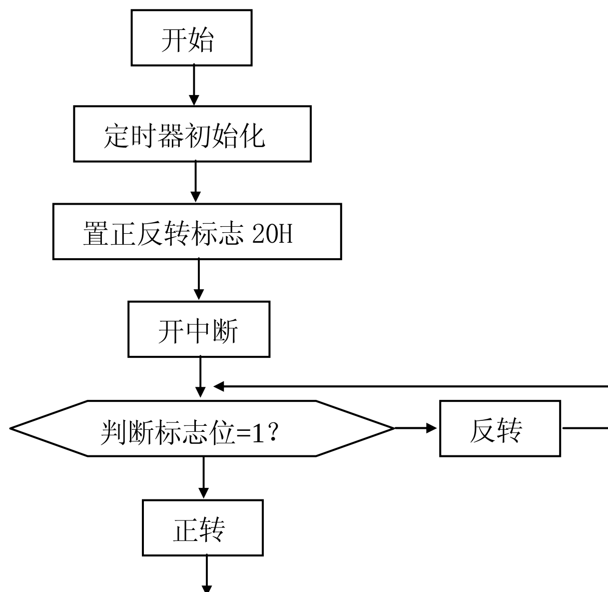
（3）存程序的目标代码，以形成文件。关了电源，卸下仿真器。

（4）连接好编程器，把目标代码烧录至 AT89C2051 芯片中。

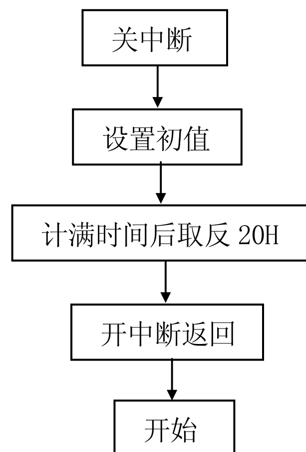
（5）然后把 AT89C2051 芯片插入锁进插座中，连接晶振源、复位电路、VCC、VSS。

（6）打开电源。步进电机和 LED 应与仿真时一样转动和闪亮。

6、程序框图：



主程序流程图



中断处理子程序框图

7、芯片引脚：

Pin	Left	Right	Pin
1	RST	VCC	20
2	(RXD)P3.0	P1.7	19
3	(TXD)P3.1	P1.6	18
4	XTAL2	P1.5	17
5	XTAL1	P1.4	16
6	(INT0)P3.2	P1.3	15
7	(INT1)P3.2	P1.2	14
8	(T0)P3.4	P1.1(AIN1)	13
9	(T1)P3.5	P1.0(AIN0)	12
10	GND	P3.7	11

89C2051

8、思考问题：

（1）使步进电机不等距转动。

9.程序清单 (MCS51\C14.ASM)

实验十五　GAL16V8 实验

1.实验目的：

学习 GAL 可编程逻辑器件的开发步骤

2.实验内容：

汽车拐弯灯控制系统

3.预备知识：

3.1　GAL 是 PAL 的第二代产品。GAL 独特的特点如下：

3.1.1　GAL 在每个输出引脚上都有输出逻辑宏单元 OLMC（Output Logic Macro Cell），允许使用者定义每个输出的结构和功能，使用户能完成任何所需要的功能。早期的可编程逻辑器件允许用户定义功能，但不允许它预先确定结构。把 GAL 与固定输出结构的可编程逻辑器件进行对比，就如同把固定结构的 PLD 与 SSI/MSI 对比一样。

3.1.2　GAL可在线用电擦洗（不是用紫外线擦洗）iSP16Z8 可擦洗一万次，其它为一百次，数据可保存 20 年。采用低功耗、高速度E^2COMS工艺制造。

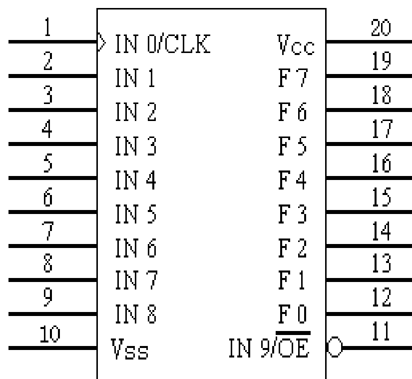
3.1.3　GAL 器件具有较高的响应速度，与 TTL 兼容。

3.1.4　GAL 器件具有电信号标签。

3.1.5　GAL 有可编程的保密位，可以防止对 GAL 的非法复制。

这些特点使 GAL 可以降低系统造价，减少产品体积和功耗，提高可靠性和稳定性，简化系统设计以及提高应用的保密性等。

3.2　GAL16V8 引脚图

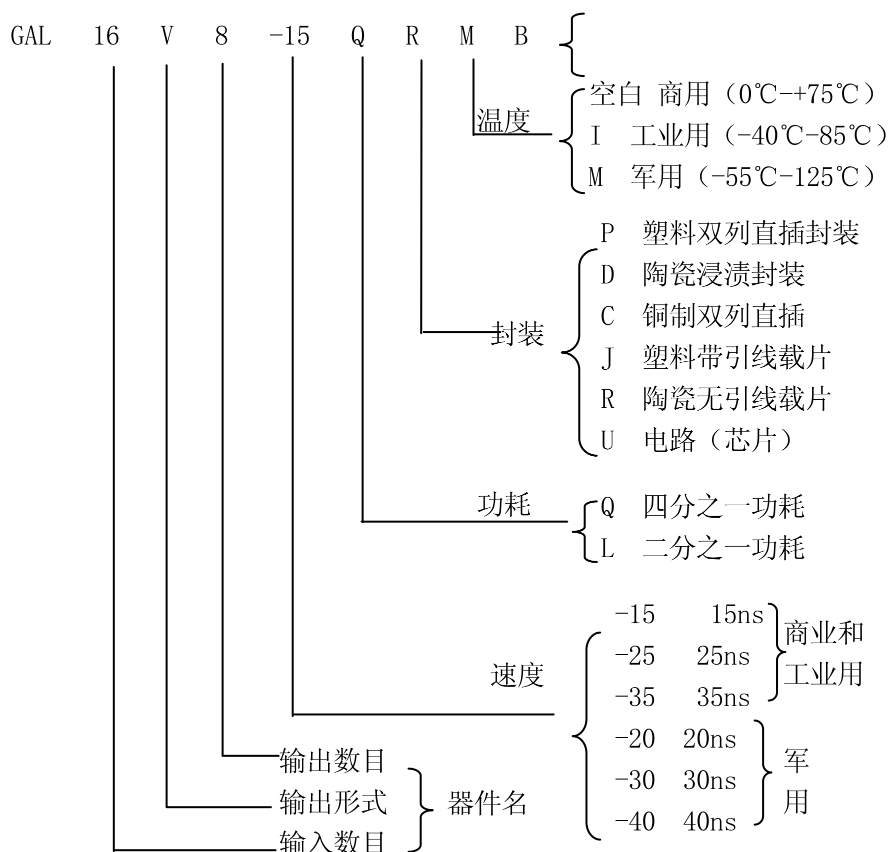


图中管脚 2-9 为八个输入缓冲器，有逻辑变量的正反两种输入。管脚 12-19 为八个输出逻辑宏单元 OLMC。管脚 1 为时钟 CK 端，管脚 11 为输出三态控制端 OE，这两个管脚有时可为输入。管脚 20 为+5V 电源端，管脚 10 为地端。

3.3　GAL 器件的命名

空白　非熔断

B　　熔断



3.4 GAL 开发步骤

(1) 根据设计要求写出真值表和逻辑表达式。在对 GAL 编程之前，首先要确定什么熔丝需要熔断以形成特定功能。事实上，所谓的编程就是将已选定的 GAL 芯片上的熔丝，按需要进行熔断或保留，它的依据便是逻辑方程式。写逻辑方程式需要一定的技术（如卡诺图），在应用这些技术之前，一般要列出真值表。这里需要说明的是，假使在设计中采用新的如“ABEL”和“CUPL”之类的软件包，它们不但能够产生熔丝图，而且还能够帮助开发逻辑方程式。在大多数情况下，它们只按真值表或者按状态图提供简单的程序，新的软件包就可以自行产生逻辑方程式和检查逻辑方程的测试向量，这样可以确保设计者得到正确无误和预期的功能.在写逻辑方程时必须注意下面两种不同的表达方式：

〈符号〉=〈表达式〉　　这种表达方式用于组合电路；

〈符号〉:=〈表达式〉　　这种表达方式用于时序电路。仅在时钟脉冲前沿时才允许输出。

(2)选择所需要芯片的型号。如选择 GAL 中的 GAL16V18、GAL39V18、GAL20V8 等。

(3)根据所选 GAL 芯片给各管脚赋以变量名。

(4)在 PC 机上用 EDLIN 或 WORDSTAR 软件准备好 PLD 文件。

(5)在 PC 机上进行编译。

A. 调用 FM . EXE；

B. 建立文本文件（LST 文件）；

C. 建立熔丝图文件（PLT 文件）；

D. 建立 JED 文件（二进制代码）。

(6)硬件编程。这时要接上 GAL 编程器，把所产生的 JED 文件烧录到 GAL 器件中。

3.5 PLD 文件的书写格式

3.5.1 FAST MAP 可解释的 GAL 器件型号是 GAL16V8 和 GAL20V8；

FAST MAP 支持以下的逻辑关系

1. “*”：表示“与”关系
2. “+”：表示“或”关系
3. “/”:表示“非”关系
4. “;”表示注释

   在每一行中“;”后面的所有 ASCII 码仅仅列入列表文件
5. “DESCRIPTION”表示文件结束。“DESCRIPTION”后的所有内容

   FASTMAP 都不予理会，仅仅写入列表文件中。

3.5.2 FAST MAP 汇编源文件的书写格式

汇编的源文件是一个标准的 ASCII 码文件，任何文本编辑如 EDLIN 、WORDSTAR 等都可以编写，FAST MAP 对于源文件书写格式有一定要求。文件的第一行要求大写字母给出 GAL 器件的型号，“PLD16V8”或“PLD20V8”，这一行不能省略；第二行给出 GAL 器件的逻辑名称，可根据需要填写，也可以省略掉，键入回车。第四行应写明设计者的姓名，同样也可以省略键入回车。但这四行一定要保留，管脚定义从第五行开始否则不能通过汇编扫描。从第五行开始是定义器件的管脚名称，管脚名称可以使用不含前面提到符号或字串，定义时只需将管脚的序号依次排列即可，而不必考虑这些名称是否写在同一行里（如果写在同一行里，两个管脚名之间应该用空格分开）。管脚定义完成之后就可以书写布尔表达式，布尔表达式中只 能 FAST MAP 所支持的 3 种关系符，根据 GAL 器件型号，每一个布尔表达式最多有 8 个“或”项，逻辑关系需要用户化简并且 FAST MAP 不支持“( )”。逻辑表过式是否正确，GAL 内部的逻辑关系及管脚排列是否合理，不但与用户的书写格式有关，而且与 GAL 的内部结构有关，因此使用者想正确使 FAST MAP 还必须通过器件手册清楚 GAL 的内部结构。当所有布尔表达式书写完毕以后，应以一个“DESCRIPTION”通知 FAST MAP 结束汇编过程，并适当追加注释，以提高汇编程序的可读性。 每一行都必须顶头开始书写。

3.6 编译软件 FM 的使用

软件编译时，调用 FM . EXE。敲入 FM 后显示菜单如下（例如文件名是 BAC.PLD）:

1. Creat Document File(source plus pinout)产生源文件和芯片管脚分配图。
2. Creat Fuse Plot File(human readable fuse map)产生可供阅读的熔丝图。
3. Creat JEDEC (programmer fuse map)产生以 JED 结尾的可写入 GAL 的文件。
4. Get A New Source File 得到新的源程序。

5. Exit Prom Fast Map 退出 Fast Map 之后问要产生哪一个文件，提示打入 1、2、……，产生所需要的文件同时存盘。

分别打入 1、2、3 在提示后再输入文件名无须后缀，回车确认后，即分别产生后缀为 PLT、JED、LST 的文件。

4. 功能分析:

汽车上有一转弯控制杆，此控制杆有三个位置：中间位置时，汽车不转向；向上时，汽车左转；向下时，汽车右转。汽车转弯时，要求左右尾灯，右头灯和仪表板上的三个指示灯相应地发出闪烁信号 。 当应急开关合上时，所有 6 个信号灯都应闪烁。汽车刹车时，两个尾灯发出不闪烁的信号。刹车时正在转弯，则相应的转弯闪烁信号不受影响。 而且当汽车转弯或应急状态下，外部信号灯和仪表指示灯的为低频闪烁； 当停靠开关合上时，外部信号灯以高频闪烁。即要有两种频率的闪烁信号输入。

5. 实验设备：

（1）G2010+实验平台　1　台　　（2）G6W 仿真器　1　台

（3）连线　若干　根　　（4）GAL16V8 芯片　1　片

（5）通用烧录器　1　台

6. 实验步骤:

（1）功能真值表：

输入信号					输出信号					
刹车 A1	应急 A2	左转 A3	右转 A4	停靠 A5	左表灯 Q1	左前灯 Q2	左后灯 Q3	右表灯 Q11	右前灯 Q22	右后灯 Q33
0	0	0	0	0	断	断	断	断	断	断
0	0	0	1	0	断	断	断	低频闪烁	低频闪烁	低频闪烁
0	0	1	0	0	低频闪烁	低频闪烁	低频闪烁	断	断	断
0	1	0	0	0	低频闪烁	低频闪烁	低频闪烁	低频闪烁	低频闪烁	低频闪烁
0	1	0	1	0	低频闪烁	低频闪烁	低频闪烁	低频闪烁	低频闪烁	低频闪烁
0	1	1	0	0	低频闪烁	低频闪烁	低频闪烁	低频闪烁	低频闪烁	低频闪烁
1	0	0	0	0	断	断	通	断	断	通
1	0	1	0	0	低频闪烁	低频闪烁	低频闪烁	断	断	通
1	0	0	1	0	断	断	通	低频闪烁	低频闪烁	低频闪烁
1	0	1	0	0	低频闪烁	低频闪烁	低频闪烁	断	断	通
1	1	0	0	0	低频闪烁	低频闪烁	通	低频闪烁	低频闪烁	通
1	1	0	1	0	低频闪烁	低频闪烁	通	低频闪烁	低频闪烁	低频闪烁
1	1	1	0	0	低频闪烁	低频闪烁	低频闪烁	低频闪烁	低频闪烁	通
×	×	×	×	×	与A_5=0 同	高频闪烁	高频闪烁	与A_5=0 同	高频闪烁	高频闪烁
×	×	×	×	×	与A_5=0 同	高频闪烁	高频闪烁	与A_5=0 同	高频闪烁	高频闪烁
×	×	×	×	×	与A_5=0 同	高频闪烁	高频闪烁	与A_5=0 同	高频闪烁	高频闪烁

（2）按下列真值表，使用 EDIT 或其它编辑器输入如下表达式：(MCS51\BUS.PLD)

PLD16V8

CONTOL LIGHT OF THE CAR

ZX2

A_1 A_2 A_3 A_4 A_5

X Y NO NO GND

NO NO Q_1 Q_2 Q_3

Q_{11} Q_{22} Q_{33} NO VCC

Q_1=A_2*x+A_3*x

Q_2=A_2*x+A_3*x+A_5*Y

Q_3=A_2*x+A_3*x+A_5*Y+/A_3*A_1

Q_{11}=A_2*x+A*x

Q_{22}=A_2*+A_4*x+A_5*Y

Q_{33}=A_1*/A_4+A_2*x+A_4*x+A_5*Y

DESCRIPTION

(3) 调用 FM 开发软件，编译以上源程序，以生成 BUS.JED 文件

(4) 连接烧录工具，把 BUS.JED 文件烧录至 GAL16V8 芯片之中去。

7. 验证结果:

GAL16V8 管脚分配图（见右）

(1) CPU8031 的P10 孔连分频器$F_入$孔。

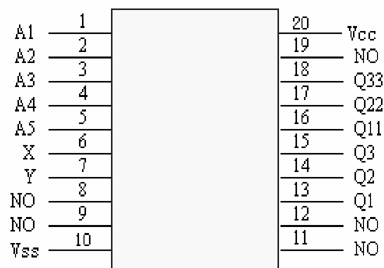
GAL 6 脚连分频器$F_入$/8 孔（低频信号）

GAL 7 脚连分频器F 入/2 孔（高频信号）

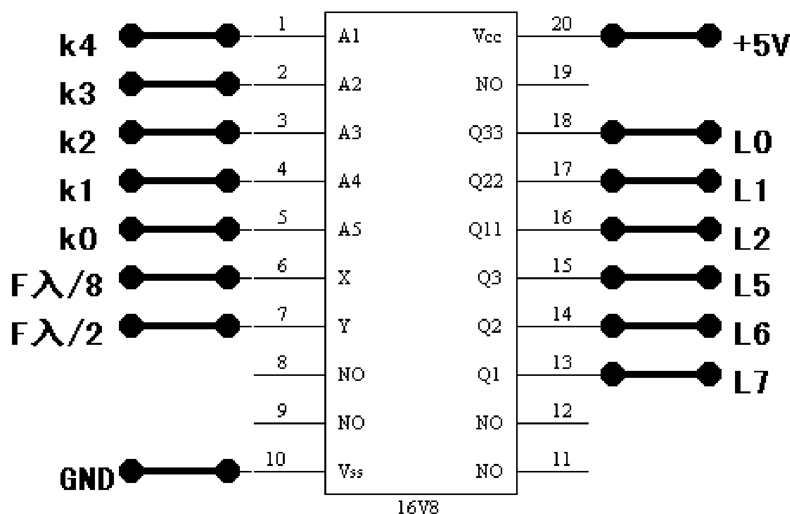
GAL 1～5 脚连 K0～K4 孔（输入）

GAL 13-15 脚连 L1-L3 孔（输出）

GAL16 孔-18 孔连L_{10}-L_{12}孔（输出）



GAL16V8 管脚分配图



（2）把实验仪与计算机相连，运行以下程序（MCS51\C15.ASM）以下产生闪烁信号

```
        ORG      0000H
        MOV  SP，#30H
 DD：   LCALL  P11
        SJMP   DD
P11：   CLR  P1.0
        MOV R2， #8H
        LCALL DELY
        SETB P1.0
        MOV R2，#8H
        LCALL DELY
        RET
DELY：  PUSH 02H
DELY1：PUSH 02H
DELY2：PUSH  02H
DELY3：PUSH  02H
DELY4：DJNZ R2，DELY4
        POP  02H
        DJNZ R2，DELY3
        POP  02H
        DJNZ R2，DELY2
        POP  02H
        DJNZ R2，DELY1
        POP  02H
        DJNZ R2，DELY
        RET
```

（3）按照真值表，拨动相应开关，来验证 LED 闪亮状态正确否。

实验十六　译码法、线选法实验

1．实验目的：

学习译码法、线选法的方法。

2．实验内容：

利用74LS138“3-8”译码器进行译码法实验，使得給予AD0809饿片选地址：CS4=2000H；利用线选法，使得給予AD0809片选地址：CS4=/A13。片选地址是低电平有效，修改AD0809实验的程序，重做该实验。

3．实验原理：

MCS51外部程序存储器一般由单片的EPROM电路组成，它独占64K程序存储器地址空间，因此不需要译码，将EPROM的片选端接地就可以了。扩展的数据存储器和I/O接口一般由多片电路组成，它们共占64KB的数据存储器地址空间，CPU是根据地址来选择RAM/IO芯片以进行信息交换的，它们的地址由地址译码的方法所确定的。

全地址译码法：对于需要扩展较多RAM/IO的系统，需采用全地址译码方法，低位地址线作为扩展电路的单元地址线（取外部电路中最多的地址线位数），对高位地址线用译码器译码，译出片选信号。常用的译码器为：

74LS139：“2-4” 译码器 ；74LS138：“3-8” 译码器 ；74LS154：“4-16” 译码器

所谓线选法就是把某一位地址线直接连到扩展电路片选端，一般片选端低电平有效，只要这一位地址线为低电平，CPU 就选中该电路进行读/写。采用线选法时，扩展 RAM/IO 的地址可以这样确定：用作片选的地址线为0，RAM电路单元或IO电路的端口地址线由CPU的寻址所确定，而其它没有用到的地址线为1。

线选法的优点是硬件简单，但由于所用的片选信号线都是高位地址线，它们的权值比较大，因此，地址空间没有被充分利用。线选法中，RAM/IO地址出现不衔接现象。

4．功能分析：

（1）　74LS138“3-8”译码器的功能表：

G1	G2A	G2B	C	B	A	/Y7	/Y6	/Y5	/Y4	/Y3	/Y2	/Y1	/Y0
1	0	0	0	0	0	1	1	1	1	1	1	1	0
1	0	0	0	0	1	1	1	1	1	1	1	0	1
1	0	0	0	1	0	1	1	1	1	1	0	1	1
1	0	0	0	1	1	1	1	1	1	0	1	1	1
1	0	0	1	0	0	1	1	1	0	1	1	1	1
1	0	0	1	0	1	1	1	0	1	1	1	1	1
1	0	0	1	1	0	1	0	1	1	1	1	1	1
1	0	0	1			0	1	1	1	1	1	1	1
其它状态			X	X	X	1	1	1	1	1	1	1	1

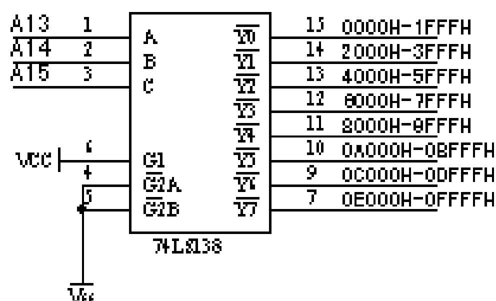
按实验内容要求：使得給予LCD的片选地址分别：CS4=2000H，可把A15、A14、A13

地址线分别接 C、B、A，于是得到的地址空间为：

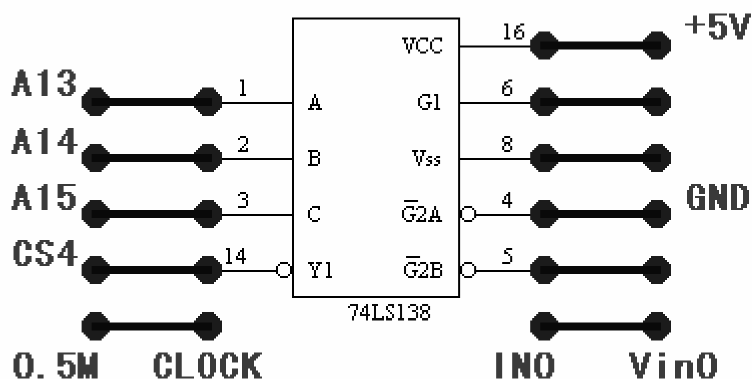
C	B	A	A12	A11	A10	A9	A8	A7-A0	译码输出	地址范围
A15	A14	A13								
0	0	0	X	X	X	X	X	X	Y0	0000H-1FFFH
0	0	1	X	X	X	X	X	X	Y1	2000H-3FFFH
0	1	0	X	X	X	X	X	X	Y2	4000H-5FFFH
0	1	1	X	X	X	X	X	X	Y3	6000H-7FFFH
1	0	0	X	X	X	X	X	X	Y4	8000H-9FFFH
1	0	1	X	X	X	X	X	X	Y5	0A000H-0BFFFH
1	1	0	X	X	X	X	X	X	Y6	0C000H-0DFFFH
1	1	1	X	X	X	X	X	X	Y7	0E000H-0FFFFH

（2）线选法：按实验内容要求，CS4=A13=0，故地址为：1101，0000，0000，0000B

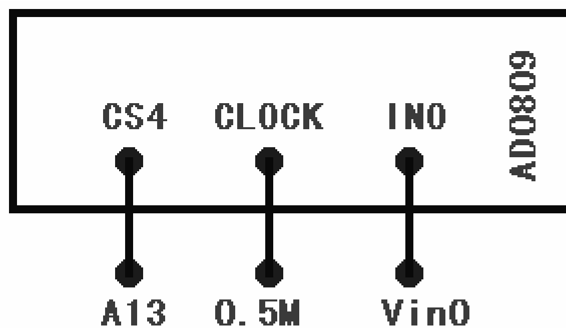
5. 接线方案：



74LS138 译码器“译码法”实验原理图



74LS138 译码器“译码法”实验接线图



“线选法”实验接线图

6．实验步骤：

（1）按“译码法”接线图案连接硬件，修该前边“AD0809”实验的程序，重做前面的实验。

（2）按“线选法”接线图案连接硬件，修该前边“AD0809”实验的程序，重做前面的实验。

7．程序清单：(MCS51\C16A.ASM) (MCS51\C16B.ASM)

第❹节 “自检式”演示实验

通过随机配带的 34 芯扁平电缆，把“自检接口 A”和“自检接口 B”连接起来，在实验平台的键盘上输入各模块“代号”，即可进行自检演示实验。

用　途：

① 可适用于初学者以“提高对单片机感性认识”为目的的实验要求。

② 实验过程中可迅速分辨是软件故障还是硬件故障。

③ 可极大地降低管理员检测或维护系统时的工作量。

方　法：

① 把 Lab51CPU 板插入 G2010+实验平台的右下角 J1、J2、J3 插座中。

② 通过随机配带的 34 芯扁平电缆，把“自检接口 A”和“自检接口 B”连接起来。

③ 打开电源，数码管应显示“P”，按[MON]键后，显示“ ’ ”，输入“代号”，即可进行自检演示实验。

④ 例：按[0]键，为“LCD 液晶显示器实验”；按[1]键，为“V/F 压力频率转换实验”

0 —— **LCD 液晶显示器实验**

显示“蓬勃远大”。

参阅 Page102：　实验七　点阵式 LCD 液晶显示屏实验

1 —— **V/F 压力频率转换实验**

调节“模拟量发生器”的电位器，数码管显示值随之变化。

参阅 Page94：　实验四　V/F 压力频率转换实验

2 —— **脉冲计数实验**

按动“脉冲源”区的带锁按钮，数码管显示值随之加 1

参阅 Page68：　实验十三　脉冲计数（定时/计数器计数功能实验）

3 —— **DA0832 转换实验**

直流电机转速随着数码管显示值加大而加大，减小而减小。

参阅 Page75：　实验十七　DA0832 转换实验

4	——	**AD0809 转换实验** 调节“模拟量发生器”的电位器，数码管显示值随之变化。范围：00-FF。 参阅 Page73：　实验十六　AD0809 转换实验
5	——	**红外线遥控实验** 需两台实验箱对做，一台键盘按值在另一台数码管显示出来。 参阅 Page114：　实验十一　红外线遥控实验
6	——	**温度测量实验** 若数码管显示的初始值不为零，则需调节“温度传感器”的电位器。 用手捏住温度传感器，数码管显示值将增大。 参阅 Page109：　实验八　温度测量实验
7	——	**力测量实验** 若数码管显示的初始值不为零，则需调节“压力传感器”的电位器。 用手压住压力传感器，数码管显示值将随之变化。 参阅 Page96：　实验五　力测量实验
8	——	**电子琴** 按“0-7”键，发出不同音调，其它键无用。 参阅 Page76：　实验十八　电子琴
9	——	**直流电机转速测量与控制实验** 在键盘上输入设定值如：40，直流电机转速将逐渐加快，在设定转速 40 左右恒速运转。 参阅 Page98：　实验六　直流电机转速测量与控制实验
A	——	**点阵 LED 广告屏实验** 显示“5” 参阅 Page112：　实验十　点阵 LED 广告屏实验
B	——	**步进电机控制** 加速跑 100 步，恒速跑 100 步，减速跑 100 步，然后反向转。 参阅 Page78：　实验十九　步进电机控制
C	——	**RS232 通讯实验** P3.0 孔连 RX1 孔，P3.1 孔连 TD1 孔,实现自发自收。 参阅 Page92：　实验三　双机通讯实验

D —— **八段码管显示实验**

在数码管上循环显示“0-F”

参阅 Page63：　　实验十一　八段码管显示实验

E —— **键盘扫描显示实验**

在数码管上显示按键值

参阅 Page65：　　实验十二　键盘扫描显示实验

F —— **扩展时钟系统（DS12887）实验**

在数码管上显示“时钟”

参阅 Page88：　　实验二　扩展时钟系统（DS12887）实验